



TUGAS AKHIR - SM 141501

DERET TAYLOR UNTUK MENYELESAIKAN MASALAH NILAI BATAS KELAS SINGULAR NONLINEAR EKSPONENSIAL PADA TEORI EKSPLOSI TERMAL

**MUNAWAROH
NRP 1211 100 010**

**Dosen Pembimbing
Dra. Sri Suprpti H., M.Si**

**JURUSAN MATEMATIKA
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember
Surabaya 2015**



FINAL PROJECT - SM 141501

***TAYLOR SERIES FOR SOLVING A CLASS OF
NONLINEAR SINGULAR EXPONENTIAL BOUNDARY
VALUE PROBLEM IN THEORY OF THERMAL
EXPLOSIONS***

MUNAWAROH
NRP 1211 100 010

Supervisor
Dra. Sri Suprpti H., M.Si

DEPARTMENT OF MATHEMATICS
Faculty of Mathematics and Natural Science
Sepuluh Nopember Institute of Technology
Surabaya 2015

LEMBAR PENGESAHAN

**DERET TAYLOR UNTUK MENYELESAIKAN
MASALAH NILAI BATAS KELAS SINGULAR
NONLINEAR EKSPONENSIAL PADA TEORI
EKSPLOSI THERMAL**

***TAYLOR SERIES FOR SOLVING A CLASS OF
NONLINEAR SINGULAR EXPONENTIAL BOUNDARY
VALUE PROBLEM IN THEORY OF THERMAL
EXPLOSIONS***

**Diajukan Untuk Memenuhi Salah Satu Syarat
Untuk memperoleh gelar Sarjana Sains
pada**

**Bidang Studi Pemodelan dan Simulasi Sistem
Program Studi S-1 Jurusan Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember**

Oleh :

**MUNAWAROH
NRP. 1211 100 010**

Menyetujui,

Dosen Pembimbing,

**Dra. Sri Suprapti H., M.Si
NIP. 19540222 198403 2 001**

**Mengetahui,
Ketua Jurusan Matematika
FMIPA ITS**

**Prof. Dr. Erna Apriliani, M.Si
NIP. 19660414 199102 2 001**

Surabaya, Januari 2015

**DERET TAYLOR UNTUK MENYELESAIKAN MASALAH
NILAI BATAS KELAS SINGULAR NONLINEAR
EKSPONENSIAL PADA TEORI EKSPLOSI TERMAL**

Nama : Munawaroh
NRP : 1211 100 010
Jurusan : Matematika FMIPA-ITS
Dosen Pembimbing : Dra. Sri Suprapti H., M.Si

Abstrak

Teori eksplosi termal didasarkan pada gagasan bahwa pemanasan progresif akan meningkatkan panas yang dilepaskan oleh reaksi sampai melebihi tingkat pelepasan panas dari daerah reaksi. Pada campuran komposisi dan tekanan tertentu yang diberikan, ledakan akan terjadi pada suhu pemanasan tertentu yang dapat ditentukan dari perhitungan pelepasan panas dan penyerapan panas. Metode deret Taylor/Taylor series method (TSM) diterapkan pada masalah nilai batas kelas singular/singular boundary value problems (SBVP) nonlinear tanpa memerlukan teknik khusus dalam menangani singularitas titik asal. Untuk masalah dengan nonlinear eksponensial, transformasi variabel diperkenalkan untuk mempercepat konvergensi dari solusi dengan mengubah nonlinear eksponensial ini menjadi nonlinier polinomial. TSM telah terbukti menjadi salah satu teknik yang berguna untuk memecahkan persamaan diferensial linear dan nonlinear biasa dengan tingkat konvergensi cepat dan kesalahan perhitungan kecil. Pendekatan Padé digunakan untuk memverifikasi bahwa solusi deret untuk masalah ini berada dalam selang batas.

Kata kunci: Masalah Nilai Batas Singular/ Singular Boundary Value Problem (SBVP), Metode Deret Taylor/Taylor Series Method (TSM), Pendekatan Padé, Teori Eksplosi Termal.



“Halaman ini sengaja dikosongkan”

**TAYLOR SERIES FOR SOLVING A CLASS OF
NONLINEAR SINGULAR EXPONENTIAL BOUNDARY
VALUE PROBLEM IN THEORY OF THE THERMAL
EXPLOSIONS**

Name : Munawaroh
NRP : 1211 100 010
Department : Mathematics FMIPA-ITS
Supervisor : Dra. Sri Suprapti H., M.Si

Abstract

Thermal explosion theory is based on the idea that progressive heating raises the rate at which heat is released by the reaction until it exceeds the rate of heat loss from the area. At a given composition of the mixture and a given pressure, explosion will occur at a specific ignition temperature that can be determined from the calculations of heat loss and heat gain. Taylor series method (TSM) is applied to a class of nonlinear singular boundary value problems (SBVP) without requiring any specific technique in handling the singularity at the origin. For the problems with exponential nonlinearity, a variable transformation is introduced to accelerate the convergence of the solution by converting this exponential nonlinearity into a polynomial nonlinearity. TSM has been proved to be one of the useful techniques to solve the linear and nonlinear ordinary differential equations (ODEs) with a fast convergence rate and small calculation error. Padé approximants are used to verify that the series solutions to these problems converge within the boundaries.

Key-words: *Thermal explosion theory, Taylor series method (TSM), singular boundary value problem (SBVP), Padé approximant.*



“Halaman ini sengaja dikosongkan”

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.,

Alhamdulillahirobbilalamiin, puji syukur penulis panjatkan kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga penulis mampu menyelesaikan Tugas Akhir yang berjudul **“Deret Taylor Untuk Menyelesaikan Masalah Nilai Batas Kelas Singular Nonlinear Eksponensial Pada Teori Eksplosi Termal”** sebagai salah satu syarat kelulusan dalam menempuh Program Sarjana Jurusan Matematika FMIPA Institut Teknologi Sepuluh Nopember (ITS) Surabaya.

Tugas akhir ini dapat terselesaikan dengan baik karena bantuan, kerja sama, dan dukungan berbagai pihak. Oleh karena itu, penulis bermaksud menyampaikan terima kasih dan penghargaan kepada :

1. Ibu Prof. Dr. Erna Apriliani, M.Si selaku Ketua Jurusan Matematika FMIPA ITS.
2. Ibu Dra. Sri Suprapti H., M.Si selaku dosen pembimbing yang telah memberikan bimbingan, pengarahan, dan motivasi dalam pengerjaan Tugas Akhir ini sehingga dapat terselesaikan dengan baik.
3. Bapak Drs. Sentot Didik Surjanto, M.Si, Dr. Hariyanto, M.Si, dan Drs. Lukman Hanafi, M.Sc selaku dosen penguji atas semua saran dan masukan yang diberikan demi perbaikan Tugas Akhir ini.
4. Bapak Drs. Chairul Imron, M.Komp selaku Koordinator Tugas Akhir Jurusan Matematika FMIPA ITS.
5. Bapak Drs. Suharmadi, Dipl. Sc, M.Phil selaku dosen penguji dan dosen wali atas masukan yang diberikan demi perbaikan Tugas Akhir ini dan arahan akademik selama

penulis menempuh perkuliahan di Jurusan Matematika ITS.

6. Bapak dan Ibu dosen serta seluruh staf Tata Usaha dan Laboratorium yang tidak bisa penulis sebutkan satu-persatu Jurusan Matematika FMIPA ITS.

Penulis menyadari bahwa Tugas Akhir ini masih terdapat kekurangan. Oleh karena itu, penulis memohon saran dan kritik dari berbagai pihak guna perbaikan Tugas Akhir ini di masa yang akan datang. Akhirnya, penulis berharap semoga Tugas Akhir ini bermanfaat bagi semua pihak.

Surabaya, Januari 2015

Penulis

Special Thank's to :

Keberhasilan terselesaikannya penulisan Tugas Akhir ini tidak lepas dari orang – orang terdekat penulis. Oleh karena itu, penulis mengucapkan terima kasih kepada :

1. Allah SWT yang telah memberikan rahmat, petunjuk, kekuatan, dan ni'mat kesabaran dalam setiap langkah kehidupan penulis dan kepada nabiulloh Muhammad SAW yang telah membimbing umatnya dari jaman kegelapan ke pada jaman yang penuh barokah ilmu ini.
2. Bapak dan ibu, kedua orang tua ku tersayang dan terkasih terimakasih atas kasih sayang, do'a, nasihat-nasihat, dan motivasinya. Semoga ni'mat Allah selalu menyertai kalian.
3. Saudara-saudaraku yang sangat kusayangi, dedek Nisa dan dedek Isna. Senyum dan tawa kalian adalah semangatku. Selalu rajin belajar dan nurut kata bapak ibu di rumah dan jangan nakal. Untuk dedek Isna sukses buat khotmil Qur'an-nya dan untuk dedek Nisa sayangku jaga ibu bapak dirumah ya ☺ ...
4. Kakak-kakaku yang kuhormati. Kak Jamik, Mas In, Mas Hat, dan Bang Mamat terimakasih kalian telah mengajarkanku arti keluarga yang sebenarnya. Sukses buat cita-cita kalian.
5. Teman-temanku *se-PASRA* (Pasukan Asrama) Mbak Hesti, Yessy, Tutut, dan Farah, aku sangat sayang kalian ☺, semoga kalian cepat menyusul.

6. Mantan *Bu Kaput* ku Bu Afifah Hasibuan terimakasih sudah membantu banyak, sudah mau susah-susah bantu mikir. Percayalah bahwa apapun yang kamu tanam ini bakal berbuah manis. Mungkin melalui orang lain pertolonganmu akan dibalas olehNya. Saudariku Ulva dan Dwi Afifah terimakasih atas cerita dan support kalian sangat menginspirasi. *Syukron katsiron....*
7. Teman-temanku se-Bojonegoro Lusi, Yongki, Veda, dan Riyani terimakasih surprise kue nya kemarin. Semoga ilmu kita barokah ☺
8. Nilam, Caca terimakasih sudah mau direpotkan *riwa-riwi* bersama layaknya 3 *serangkai*. Semoga ilmu kita barokah dan bermanfaat karena sungguh ilmu yang tidak diamalkan itu bagai pohon yang tak berbuah.
9. Seluruh keluarga besar HIMATIKA ITS atas dukungan dan semangat yang diberikan kepada penulis
10. Teman-temanku angkatan 2011 terimakasih kalian adalah keluarga baruku selama ini. Senang sekali rasanya berada diantara orang-orang pilihan dan hebat seperti kalian.
11. Semua pihak yang tak bisa penulis sebutkan satu-persatu, terima kasih telah membantu sampai terselesaikannya Tugas Akhir ini.

DAFTAR ISI

HALAMAN JUDUL	i
LEMBAR PENGESAHAN	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
DAFTAR SIMBOL	xxi
BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan	3
1.5 Manfaat	4
1.6 Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA	
2.1 Persamaan Diferensial Tak Linear Tingkat Dua	7
2.2 Metode Dekomposisi Adomian	8
2.3 Metode Transformasi Diferensial	10
2.4 Metode Deret Taylor	12
2.5 Pendekatan Padé	13
BAB III METODOLOGI PENELITIAN	
3.1 Studi Literatur	15
3.2 Tahap Menurunkan dan Mengkaji Model Matematika Pada Teori Eksplasi Termal	15
3.3 Tahap Mencari Solusi Eksak Model	15
3.4 Tahap Mencari Solusi Numerik Model	16
3.5 Tahap Simulasi Numerik Model Menggunakan <i>Matlab 2010a</i>	16

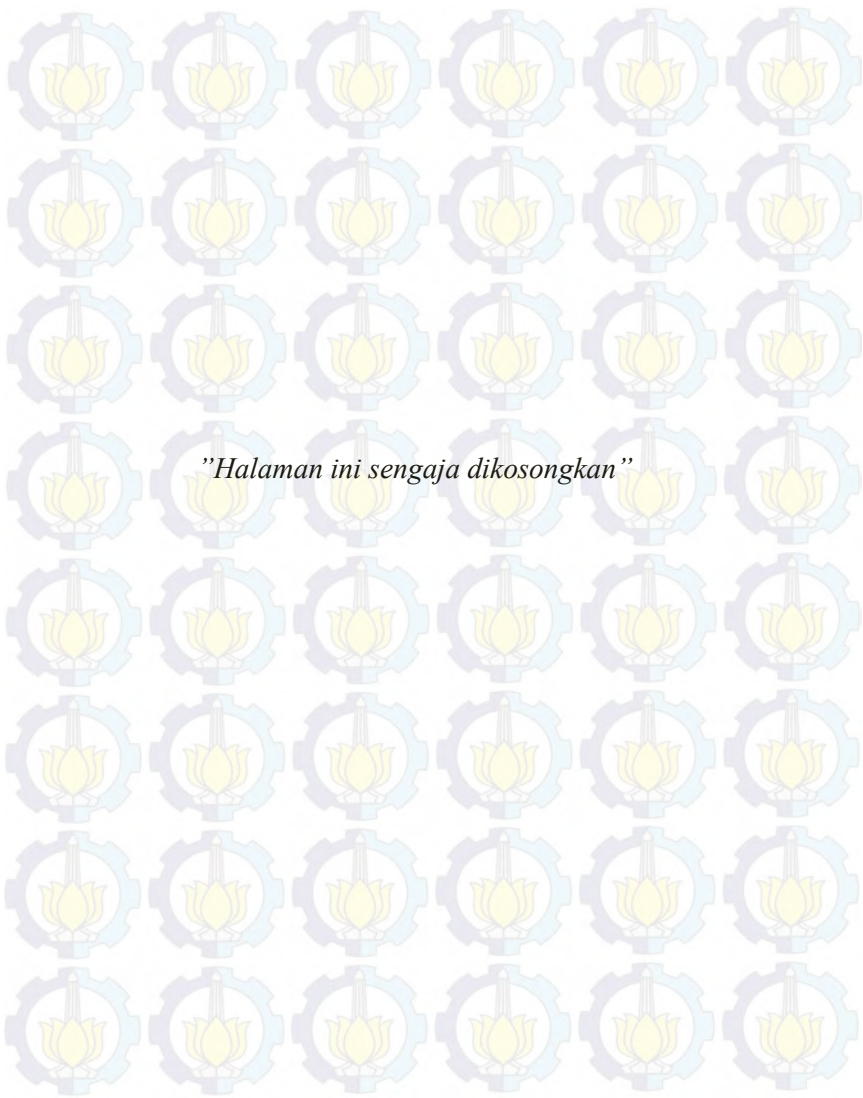
3.6	Kesimpulan dan Saran	16
3.7	Diagram Alir	16
BAB IV ANALISIS DAN PEMBAHASAN		
4.1	Definisi Dan Proses Terjadinya Eksplosi Termal	19
4.2	Penurunan Model Matematika Bentuk Eksplosi Termal	21
4.3	Solusi Eksak Model Matematika Pada Teori Eksplosi Termal	26
4.4	Solusi Numerik Model Matematika Pada Teori Eksplosi Termal Menggunakan Metode Deret Taylor	30
BAB V KESIMPULAN DAN SARAN		
5.1	Kesimpulan	47
5.2	Saran	48
DAFTAR PUSTAKA		49
LAMPIRAN		
A.	Listing Dan Running Program Mencari Nilai A saat $M = 20$	53
B.	Listing Dan Running Program Mencari Nilai A saat $M = 40$	55
C.	Listing Dan Running Program Mencari Nilai A saat $M = 60$	59
D.	Desain GUI untuk simulasi (tampilan awal dan tampilan utama)	65
E.	Listing Program M-file dengan judul Solusi_MNB_Singular.m	69
F.	Biodata Penulis	83

DAFTAR TABEL

Tabel 2.1 Beberapa sifat dari metode transformasi diferensial	11
--	----

Tabel 4.1 Fungsi Transformasi dari persamaan diferensial pada teori eksplosi termal.....	33
---	----

Tabel 4.2 Maksimum eror yang dicapai.....	45
--	----



"Halaman ini sengaja dikosongkan"

DAFTAR GAMBAR

Gambar 3.1 Diagram Alir Penelitian 17

Gambar 4.1 Plot fluks termal terhadap temperatur 20

Gambar 4.2 Reaktor Silinder Dengan Aliran Radial..... 23

Gambar 4.3 Singularitas di bidang kompleks x dengan nilai A bernilai 0.07900857355822, 0.07900857355927 dan 0.07900857355927 masing-masing untuk $M=20, 40, \text{ dan } 60$ 43

Gambar 4.3 Singularitas di bidang kompleks x dengan nilai A bernilai

- 0.027958791937068318689303963249272 i + 0.74840537855354747157484168762119
- 0.26488823999378558276401828767185 I + 1.0785841597402998260069158468162 dan,
- 5.3695481056208137856823282397704 - 13.870916256491247126448395510149 i masing-masing untuk $M=20, 40, \text{ dan } 60$ 44



“Halaman ini sengaja dikosongkan”

DAFTAR SIMBOL

θ, y adalah temperatur atau suhu yang tidak berdimensi,

η adalah tingkat pembakaran yang tidak berdimensi ,

τ adalah waktu yang tidak berdimensi,

ε dan β adalah parameter kecil yang positif,

δ adalah kriteria Frank-Kamenetsky,

A_n adalah polinomial Adomian dari bentuk nonlinear e^y

$U(k)$ adalah koefisien Taylor ke-k

DAFTAR PUSTAKA

- [1] R.K. Pandey, (1997). **“A finite difference method for a class of singular two-point boundary value problems arising in physiology”**, Int. J. Comput. Math. Vol. 65 pp:131–140.
- [2] R.K. Pandey, A.K. Singh, (2004). **“On the convergence of a finite difference method for a class of singular boundary value problems arising in physiology”**, J. Comput. Appl. Math. Vol. 166 pp: 553–564.
- [3] R.K. Pandey, A.K. Singh, (2009). **“ the convergence of a fourth-order method for a class of singular boundary value problems, J. Comput”**. Appl. Math. Vol. 224 pp: 734–742.
- [4] S.M. El-Sayed, (1998). **“Multi-integral methods for nonlinear boundary-value problems”**, Int. J. Comput. Math. Vol. 72 pp: 259–265.
- [5] S.M. El-Sayed, (2002). **“Integral methods for computing solutions of a class of singular two-point boundary value problems”**, Appl. Math. Comput. Vol. 130 pp: 235–241
- [6] A.S.V. Ravi Kanth, V. Bhattacharya, (2006). **“Cubic spline for a class of nonlinear singular boundary value problems arising in physiology”**, Appl. Math. Comput. Pp: 768–774.
- [7] J. Rashidinia, R. Mohammadi, R. Jalilian, (2007). **“The numerical solution of nonlinear singular boundary value problems arising in physiology”**, Appl. Math. Comput. Vol. 185 pp: 360–367.
- [8] H. Caglar, N. Caglar, M. Ozer, (2009). **“B-spline solution of nonlinear singular boundary value problems arising in physiology”**, Chaos, Solitons Fractals Vol. 39 pp: 1324–1334.
- [9] S.A. Khuri, A. Sayfy, (2010) **“A novel approach for the solution of a class of singular boundary value problems arising in physiology”**, Math. Comput. Model. Vol. 52 pp: 626–636.
- [10] M. Kumar, N. Singh, (2010). **“Modified Adomian decomposition method and computer implementation for solving singular boundary value problems arising in**

- various physical problems**", Comput. Chem. Eng. Vol. 52 pp: 1750–1760.
- [11] A.S.V. Ravi Kanth, K. Aruna, (2010). "**He's variational iteration method for treating nonlinear singular boundary value problems**", Comput. Math. Appl. Vol. 60 pp: 821–829.
- [12] A.M. Wazwaz, (2011). "**The variational iteration method for solving nonlinear singular boundary value problems arising in various physical models**", Commun. Nonlinear Sci. Numer. Simul. Vol.16 pp: 3881–3886.
- [13] Ebaid, Abd Elhalim, 2010. "**Exact Solutions for a Class of Nonlinear Singular Two-Point Boundary Value Problems: The Decompositiod Method**" Z. Naturforsch. Vol. 65a pp: (145 – 150).
- [14] Gübeş, Murat, "**Applications of Differential Transform Method for ENSO Model with compared ADM and VIM**" Department of Mathematics, Karamanoğlu Mehmetbey University, Karaman/TÜRKİYE.
- [15] Zhou, J.K, (1986). "**Differential Transformation and Its Applications for Electrical Circuits**". Huazhong University Press, Wuhan (in Chinese)
- [16] Chang, Shih-Hsiang. (2014). "**Applied Mathematics and Computation: Taylor series method for solving a class of nonlinear singular boundary value problems arising in applied science**". Vol. 235 pp: 110–117.
- [17] Kallirath, Josef, (2002). "**On Rational Function Technique And Padé Approximants**". Jerman.
- [18] Anonim, (2014). "Possibilities Arising From The Heat Loss And Production" dalam <http://www.leads.ac.uk/fuel/tutorial/semenov/possibilities.html> (diakses tanggal 27 September 2014).
- [19] Gorelov, G.N. dan Sobolev, V.A, "**Mathematical Modelling Of Critical Condition In The Thermal Explosion Problem**", Samara State University, Russia.
- [20] M. Strauss, T.A. Ring, H.K. Bowen, (1987) . "**Osmotic pressure for concentrated suspensions of polydisperse**

particles with thick double layers", J. Colloid Interface Sci. Vol. 118 (2) pp: 326–334.

- [21] Hasan, I. H. Abdel-Halim dan Ertürk, Vedat Suat, (2009) : **"Solutions of Different Types of the linear and Non-linear Higher-Order Boundary Value Problems by Differential Transformation Method"**, Europe journal of pure and applied mathematics Vol. 2, No. 3, pp: (426-447).
- [22] Idrees, Muhammad dkk, (2013) **" Exact Solution for a Class of Stiff Systems by Differential Transform Method"** Applied Mathematics, Vol. 4, pp: 440-444
- [23] Moon, Saurabh Dilip dkk, (2013). **"Solution of Non-Linear Differential Equations by Using Differential Transformn Method"** International Journal of Mathematics and Statistics Invention (IJMSI), vol 2, pp: (78-82).
- [24] Sista, Sivaji Ganesh, **"Ordinary Differential Equations"** MA 417. India



LAMPIRAN F

BIODATA PENULIS



Penulis bernama Munawaroh, lahir di Bojonegoro pada tanggal 24 Desember 1992. Penulis merupakan anak pertama dari tiga bersaudara dari pasangan Rosul dan Sarimah. Penulis telah menempuh pendidikan formal yaitu TK Dharma Wanita Talok-Kalitidu-Bojonegoro, SD Negeri Talok-Kalitidu-Bojonegoro, SMP Negeri 1 Kalitidu-Bojonegoro, dan SMA Negeri 1 Kalitidu-Bojonegoro.

Setelah lulus dari SMA penulis melanjutkan pendidikan ke jenjang Strata 1 di Jurusan Matematika ITS melalui jalur SNMPTN Undangan pada tahun 2011 dan terdaftar dengan NRP 1211 100 010. Di Jurusan Matematika ini, penulis mengambil bidang minat Pemodelan dan Simulasi Sistem. Selain itu, penulis juga aktif di beberapa organisasi intra kampus diantaranya: Himpunan Mahasiswa Matematika (HIMATIKA ITS) sebagai staf Departemen Kesejahteraan Mahasiswa pada periode 2012-2014 dan Lembaga Dakwah Jurusan Matematika ITS “ Ibnu Muqlah” sebagai staf Keputrian pada periode 2012-2014

Kritik, saran, pertanyaan, dan informasi lebih lanjut mengenai Tugas Akhir ini dapat dikirimkan kepada penulis melalui email: munawa24@gmail.com. Terimakasih ☺

BAB I

PENDAHULUAN

Pada bab ini akan dijelaskan mengenai hal-hal yang melatarbelakangi dari permasalahan yang dibahas pada Tugas Akhir ini. Kemudian permasalahan tersebut disusun ke dalam suatu rumusan masalah. Selanjutnya untuk menyelesaikan permasalahan pada Tugas Akhir ini diberikan juga batasan masalah guna untuk mendapatkan tujuan yang diharapkan dan manfaat yang akan diperoleh. Adapun sistematika penulisan Tugas Akhir ini akan diuraikan pada akhir bab ini.

1.1 Latar Belakang

Perkembangan suatu ilmu pengetahuan banyak memegang peranan penting dalam perkembangan suatu teknologi. Tanpa ilmu pengetahuan, teknologi akan sulit bisa berkembang dengan cepat. Matematika bahasa simbol yang bersifat universal sangat erat hubungannya dengan kehidupan nyata. Kenyataan membuktikan bahwa untuk menyelesaikan masalah-masalah kehidupan nyata dibutuhkan metode-metode matematika.

Pada umumnya berbagai masalah nilai batas singular (BVPS) terjadi dalam ilmu alam (sains) dan teknik, memiliki non-linear eksponensial dalam bentuk

$$f(x, y) = g(x)h(e^y)$$

Model eksponensial sering ditemukan pada kasus pertumbuhan dan peluruhan. Sebagai contoh pertumbuhan uang yang diinvestasikan dalam sebuah bank, pertumbuhan suatu zat radioaktif, penambahan penduduk dalam suatu daerah dan lainnya.. Model eksponensial ini terjadi karena jumlah pertumbuhan yang bertambah atau meningkat secara drastis selama kurun waktu tertentu. Selain pada pertumbuhan manusia, model eksponensial juga terjadi pada pertumbuhan bakteri atau virus dimana pertumbuhan meningkat sebanyak 2^t dimana t adalah waktu dalam satuan detik. Sedangkan model peluruhan

eksponensial erat hubungannya dengan usia dari suatu fosil berdasarkan peluruhan radioaktif.

Masalah klasik yang sering dihadapi dalam menyelesaikan bentuk persamaan diferensial adalah menganalisis titik-titik tetap dan titik-titik kestabilan dari persamaan diferensial linear maupun nonlinear berdasarkan syarat awal dan syarat batas yang telah ditentukan.

Salah satu masalah nilai batas singular (BVPS) nonlinear eksponensial adalah pada “Teori Eksplosi Termal”. Istilah ini diambil untuk mengacu pada tahap awal perilaku campuran yang mudah terbakar karena suhu yang mulai meningkat dan bersaing dalam proses fisika dan kimia.

Sebelumnya sudah dilakukan penelitian untuk menyelesaikan masalah nilai batas singular ini. Berbagai metode numerik seperti metode beda hingga [1-3], metode integral iterasi [4,5], metode *cubic spline* [6,7], metode *B-spline* [8] dan metode dekomposisi-spline campuran [9] telah diterapkan untuk menyelesaikan BVPS dengan berbagai fungsi nonlinear $f(x, y)$ dan hasil yang dihasilkan bervariasi. Keuntungan dari metode iteratif baru-baru ini dikembangkan atas metode numerik tersebut adalah dapat memberikan pendekatan yang sangat akurat atau bahkan solusi yang tepat walaupun dengan sedikit pekerjaan komputasi dan tanpa memerlukan teknik khusus dalam menangani singularitas pada titik asal. Operator baru diusulkan untuk memberikan solusi eksak masalah ini adalah penggunaan metode dekomposisi Adomian [10] yang dimodifikasi untuk memberikan hasil yang lebih baik dan akurat di persekitaran batas yang tepat. Namun, metode iterasi variasional [11,12] tersebut gagal untuk memecahkan jenis BVPS dengan fungsi eksponensial $f(x, y)$.

Berdasarkan latar belakang yang telah dijelaskan, pada Tugas Akhir ini, akan digunakan metode deret Taylor (TSM) untuk memecahkan masalah nilai batas kelas singular nonlinear eksponensial pada “Teori Eksplosi Termal”. Tidak ada teknik khusus yang diperlukan dalam menangani singularitas pada $x = 0$ dan titik batas yang tepat pada $x = 1$ dengan cara mengkonversi

nonlinear eksponensial ini menjadi nonlinear polinomial. Kemudian memverifikasi bahwa solusi deret untuk masalah ini konvergen dalam $x \in (0,1]$ menggunakan pendekatan Padé. Selain itu juga menganalisis segala keerroran/galat yang terjadi, sehingga nantinya keadaan “Eksplasi Termal” ini dapat dihindari atau bahkan dicegah. Semua perhitungan dilakukan pada komputer/laptop menggunakan software *Matlab 2010a*.

1.2 Rumusan Masalah

Berkaitan dengan latar belakang yang ada, permasalahan yang dibahas pada Tugas Akhir ini antara lain :

1. Bagaimana menyelesaikan masalah nilai batas kelas singular nonlinear eksponensial pada “Teori Eksplasi Termal” ?
2. Bagaimana mentukan titik titik stabilitas berdasarkan diagram fasenya (grafik) yang diperoleh dari perhitungan software *Matlab 2010a* ?
3. Bagaimana menentukan nilai galat/eror yang memenuhi sehingga keadaan eksplasi termal dapat dihindari ?

1.3 Batasan Masalah

Batasan masalah yang diberikan dalam Tugas Akhir ini diantaranya sebagai berikut :

1. Deret Taylor konvergen pada interval $(0,1]$
2. Deret Taylor analitik pada interval $(0,1]$.
3. Software yang digunakan adalah *Matlab 2010a*.

1.4 Tujuan

Berdasarkan permasalahan dalam rumusan masalah yang telah dijelaskan di atas, maka tujuan dari Tugas Akhir ini adalah :

1. Menyelesaikan masalah nilai batas kelas singular nonlinear eksponensial pada “Teori Eksplasi Termal”
2. Menentukan titik-titik stabilitas berdasarkan diagram fasenya (grafik) yang diperoleh dari perhitungan software *Matlab 2010a*

3. Menentukan nilai eror atau galat yang memenuhi sehingga keadaan eksplosi termal dapat dihindari.

1.5 Manfaat

Manfaat dari penulisan Tugas Akhir ini adalah sebagai rujukan (acuan) bagaimana cara untuk menghindari keadaan eksplosi termal selama reaksi kimia berlangsung bagi para pengguna.

1.6 Sistematika Penulisan

Penulisan Tugas Akhir ini secara keseluruhan disusun atas lima bab dan lampiran. Secara garis besar masing-masing bab akan membahas hal-hal sebagai berikut :

BAB I PENDAHULUAN

Bab ini berisi tentang gambaran umum dari penulisan Tugas Akhir ini yaitu meliputi latar belakang permasalahan, perumusan masalah, batasan masalah, tujuan dan manfaat serta sistematika penulisan.

BAB II TINJAUAN PUSTAKA

Bab ini berisi tentang dasar teori dan materi pendukung yang digunakan penulis dalam mengerjakan Tugas Akhir ini, antara lain definisi dan proses terjadinya eksplosi termal, penurunan model matematika bentuk eksplosi termal, persamaan diferensial tak linear tingkat dua, metode transformasi diferensial, metode deret Taylor, dan pendekatan Padé.

BAB III METODE PENELITIAN

Bab ini menjelaskan tentang alur kerja (tahapan-tahapan) dan metode yang digunakan penulis dalam mengerjakan Tugas Akhir ini.

BAB IV ANALISIS DAN PEMBAHASAN

Bab ini menyajikan formulasi dan penyelesaian berupa solusi eksak dan solusi numerik dari persamaan diferensial yang berlaku pada teori eksplosif termal serta penjelasan mengenai hasil simulasi yang diperoleh.

BAB V PENUTUP

Bab ini berisi kesimpulan berdasarkan pembahasan sebelumnya dan saran guna pengembangan penelitian sebelumnya.

LAMPIRAN



BAB II TINJAUAN PUSTAKA

Pada bab ini akan dijelaskan mengenai tinjauan pustaka yang menjadi landasan atau dasar teori dan materi pendukung lainnya antara lain persamaan diferensial tak linear tingkat dua, metode dekomposisi Adomian, metode transformasi diferensial, metode deret Taylor, dan pendekatan Padé guna untuk mendukung penyelesaian dan penyusunan Tugas Akhir ini.

2.1 Persamaan Diferensial Tak Linear Tingkat Dua [24]

Persamaan diferensial adalah suatu bentuk persamaan yang memuat derivatif (turunan) satu atau lebih variabel tak bebas terhadap satu atau lebih variabel bebas suatu fungsi.

Persamaan diferensial biasa linear order 2 dapat dituliskan sebagai

$$a_0(x) \frac{d^2y}{dx^2} + a_1(x) \frac{dy}{dx} + a_2(x)y = b(x)$$

dengan $a_0 \neq 0$ dan a_n adalah bentuk polinomial dengan derajat terendah adalah 0.

Persamaan diferensial biasa non linear jika persamaan diferensial tersebut tak linear dan terdapat perkalian variabel bergantung dengan turunannya.

Contoh:

$$\left(\frac{d^2y}{dx^2}\right)^2 + x^3 \frac{dy}{dx} = xe^x$$

Untuk menyelesaikan persamaan diferensial nonlinear yang tidak bisa diselesaikan dengan metode yang biasa-biasa saja, banyak diperkenalkan metode-metode baru dengan mentransformasi fungsi asli sehingga memudahkan untuk mencari solusi penyelesaian dari persamaan diferensial yang diberikan. Masalah yang sering muncul dalam mencari solusi dari sebuah persamaan diferensial adalah adanya titik singularitas pada batas-batas yang diberikan sehingga perlu adanya teknik khusus untuk mengatasi singularitas pada batas-batas yang diberikan.

2.2 Metode Dekomposisi Adomian [13]

Beberapa masalah yang paling umum dalam ilmu terapan dan teknik biasanya dirumuskan sebagai masalah nilai batas singular pada dua titik berbentuk sebagai berikut

$$(x^\alpha y')' = f(x, y), \quad 0 < x \leq 1 \quad (2.1)$$

dengan

$$\lim_{x \rightarrow 0} y(x) = A \text{ dan } y(1) = B \quad (2.2)$$

Didefinisikan

$$L_{xx}(y) = f(x, y)$$

dengan L_{xx} didefinisikan dengan

$$L_{xx}[\cdot] = \frac{d}{dx} \left(x^\alpha \frac{d}{dx} [\cdot] \right) \quad (2.3)$$

Invers dari L_{xx}^{-1} didefinisikan oleh

$$L_{xx}^{-1}[\cdot] = \int_0^x \left(x^{-\alpha} \int_0^x [\cdot] dx \right) dx \quad (2.4)$$

Operasikan L_{xx}^{-1} pada persamaan (2.1) dan (2.2) menjadi berikut

$$y = y(0) + \int_0^x \left(x^{-\alpha} \int_0^x f(x, y) dx \right) dx \quad (2.5)$$

Perhatikan bahwa $y(0) = A$, syarat cukup untuk mendapatkan penyelesaian dan syarat lainnya ketika $y(1) = B$, dapat digunakan untuk menunjukkan bahwa solusi yang diperoleh

ini sesuai dengan syarat batas yang sudah diberikan. Asumsikan bahwa $f(x, y) = r(x)g(y)$, sehingga persamaan (2.5) menjadi

$$y = y(0) + \int_0^x \left(x^{-\alpha} \int_0^x r(x)g(y) dx \right) dx \quad (2.6)$$

Metode dekomposisi Adomian ini didasarkan pada dekomposisi y dan bentuk nonlinear $g(y)$ sebagai

$$y = \sum_{n=0}^{\infty} y_n, \quad g(y) = \sum_{n=0}^{\infty} A_n(y_1, y_2, y_3, \dots, y_n) \quad (2.7)$$

Dengan A_n adalah pembentuk polinomial Adomian yang khusus untuk non-linear tertentu dan dapat ditentukan dari rumus

$$A_n = \frac{1}{n!} \left[\frac{d^n}{d\lambda^n} g \left(\sum_{i=0}^{\infty} \lambda^i y_i \right) \right], \quad n \geq 0 \quad (2.8)$$

Substitusikan persamaan (2.7) ke persamaan (2.6) diperoleh

$$\sum_{n=0}^{\infty} y_n = y(0) + \sum_{n=0}^{\infty} \int_0^x \left(x^{-\alpha} \int_0^x r(x)A_n dx \right) dx \quad (2.9)$$

Menurut standar metode dekomposisi Adomian, solusi dan penyelesaian dapat dihitung dengan menggunakan relasi perulangan

$$y_0 = y(0) \quad (2.10)$$

$$y_{n+1} = \int_0^x \left(x^{-\alpha} \int_0^x [r(x)A_n] dx \right) \quad n \geq 0$$

Selanjutnya, persamaan (2.10) sangat efektif dan cocok dalam memperoleh solusi yang tepat untuk berbagai masalah nilai batas kelas singular nonlinear pada 2 titik dalam bentuk yang diberikan oleh persamaan (2.1) dan (2.2).

2.3 Metode Transformasi Diferensial [14]

Metode transformasi diferensial adalah metode numerik berdasarkan ekspansi Taylor. Metode ini mencoba untuk menemukan koefisien ekspansi serangkaian fungsi yang tidak diketahui dengan menggunakan kondisi awal pada permasalahan. Konsep metode transformasi diferensial pertama kali diusulkan oleh Zhou [15].

Definisi 1

Transformasi diferensial satu dimensi dari fungsi $y(x)$ pada titik $x = x_0$ didefinisikan

$$Y(k) = \frac{1}{k!} \left. \frac{d^k y(x)}{dx^k} \right|_{x=x_0} \quad (2.11)$$

dengan $y(x)$ adalah fungsi asli dan $Y(k)$ adalah fungsi transformasi

Definisi 2

Invers transformasi diferensial dari $Y(k)$ didefinisikan sebagai berikut

$$y(x) = \sum_{k=0}^{\infty} Y(k)(x - x_0)^k \quad (2.12)$$

Dari (2.11) dan (2.12) dapat ditulis

$$y(x) = \sum_{k=0}^{\infty} \frac{1}{k!} \left. \frac{d^k y(x)}{dx^k} \right|_{x=x_0} (x - x_0)^k \quad (2.13)$$

Terkait definisi di atas disajikan beberapa sifat dasar DTM (metode deret Taylor) sebagai berikut :

Tabel 2.1. Beberapa sifat dari metode transformasi diferensial

<i>Fungsi Asli</i>	<i>Fungsi Transformasi</i>
$f(x) = g(x) \pm h(x)$	$F(k) = G(k) \pm H(k)$
$f(x) = cg(x)$	$F(k) = c G(k)$
$f(x) = \frac{d^n g(x)}{dx^n}$ <i>Contohnya</i> $f(x) = \frac{d^2 g(x)}{dx^2}$ $f(x) = \frac{dg(x)}{dx}$	$F(k) = \frac{(k+n)!}{k!} G(k+n)$ $F(k) = \frac{(k+2)!}{k!} G(k+2)$ $= (k+1)(k+2) G(k+2)$ $F(k) = \frac{(k+1)!}{k!} G(k+1)$ $= (k+1) G(k+1)$
$f(x) = g(x)h(x)$	$F(k) = G(k) \otimes H(k)$ $= \sum_{r=0}^k G(r) H(k-r)$
$f(x) = g(x)h(x)v(x)$	$F(k) = G(k) \otimes H(k) \otimes V(k)$ $= \sum_{r=0}^k \sum_{t=0}^{k-r} G(r) H(t) V(k-r+t)$
$f(x) = g(x) \frac{d^2 h(x)}{dx^2}$	$F(k) = G(k) \otimes [(k+1)(k+2) H(k+2)]$ $= \sum_{r=0}^k (k-r+1)(k-r+2) G(r) H(k-r+2)$
$f(x) = ax^m, a \in \mathbb{R}$	$F(k) = a \delta(k-m) = \begin{cases} a, & k=m \\ 0, & k \neq m \end{cases}$

Keterangan :

\otimes adalah simbol konvolusi yang artinya cara untuk mengkombinasikan dua buah deret angka yang menghasilkan deret angka yang baru

δ adalah simbol delta Kronecker

2.4 Metode Deret Taylor [16]

Ekspansi deret Taylor dari fungsi $y(x)$ analitik pada persekitaran $x = 0$ diberikan sebagai berikut

$$y(x) = \sum_{k=0}^{\infty} \frac{d^k y(x)}{dx^k} \bigg|_{x=0} \frac{x^k}{k!} = \sum_{k=0}^{\infty} Y(k) x^k \quad (2.14)$$

dimana $y^k(x) = \frac{d^k y(x)}{dx^k}$ dan $Y(k)$ disebut koefisien Taylor ke- k dan didefinisikan sebagai

$$Y(k) = \frac{1}{k!} \frac{d^k y(x)}{dx^k} \bigg|_{x=0} \quad (2.15)$$

Fungsi $y(x)$ dapat dinyatakan sebagai $y(x) = y_M(x) + R_M$, dimana $y_M(x)$ adalah aproksimasi orde- M yang diberikan oleh

$$y_M(x) = \sum_{k=0}^M Y(k) x^k \quad (2.16)$$

dan R_M adalah rumus sisanya diberikan oleh

$$R_M = \frac{1}{(M+1)!} \frac{d^{M+1} y(x)}{dx^{M+1}} \bigg|_{x=x_1} x^{M+1} \quad (2.17)$$

untuk beberapa x_1 pada $0 < x_1 < x$. Jika turunan ke $(M+1)$ dari $y(x)$ terbatas dalam $x \in (0,1]$.

$$\left| \frac{d^{M+1}y(x)}{dx^{M+1}} \right| \leq K \quad (2.18)$$

untuk konstanta K non-negatif tertentu, maka error maksimum untuk $y_M(x)$ dalam interval ini dapat dievaluasi berdasarkan rumus sisa seperti ini

$$e_{max} = \frac{K}{(M+1)!}$$

2.5 Pendekatan Padé [17]

Pendekatan Padé adalah pendekatan yang digunakan untuk mendekati solusi penyelesaian dari sebuah persamaan diferensial yang berbentuk fungsi rasional dan sering digunakan untuk memverifikasi solusi dari sebuah deret. Bentuk umum dari Pendekatan Padé adalah :

Misalnya diberikan fungsi analitik $y(x)$ dengan ekspansi Taylor di persekitaran $x = 0$

$$y(x) = \sum_{k=0}^{\infty} y_k x^k, \quad 0 < x \leq X \quad (2.19)$$

Pendekatan Padé untuk fungsi $y(x)$ dengan order $[L, M]$ didefinisikan sebagai berikut

$$[L/M]_y(x) = \frac{p_0 + p_1x + \dots + p_Lx^L}{q_0 + q_1x + \dots + q_Mx^M} \quad (2.20)$$

dengan $q_0 = 1$, dan pembilang dan penyebut tidak memiliki faktor umum. Sehingga pembilang dan penyebut yang dibangun $y(x)$ dan $[L/M]_y(x)$ membentuk fungsi rasional dan turunannya ada pada saat $x = 0$ sampai dengan $L + M$ yaitu

$$y(x) - [L/M]_y(x) = O(x^{L+M+1}) \quad (2.21)$$

Dari persamaan di atas diperoleh

$$y(x) \sum_{k=0}^M q_k x^k - \sum_{k=0}^L p_k x^k = O(x^{L+M+1}) \quad (2.22)$$

BAB III

METODOLOGI PENELITIAN

Pada bab ini diuraikan langkah-langkah sistematis yang dilakukan dalam proses pengerjaan Tugas Akhir. Metode penelitian dalam Tugas Akhir ini terdiri atas enam tahap, antara lain: studi literatur, tahap menurunkan dan mengkaji model matematika pada teori eksplosi termal, tahap mencari solusi/ penyelesaian eksak model, tahap mencari solusi numerik model, tahap simulasi numerik model menggunakan software *Matlab 2010a*, dan penarikan kesimpulan dan saran.

3.1 Studi Literatur

Pada tahap ini dilakukan analisis model dan identifikasi permasalahan dengan mencari dan mempelajari literatur-literatur yang terkait seperti jurnal, paper, dan buku-buku maupun artikel terkait dari internet yang berhubungan dengan model matematika eksplosi termal dan solusi penyelesaian masalah nilai batas kelas singular nonlinear pada teori eksplosi termal menggunakan metode deret Taylor.

3.2 Tahap Menurunkan dan Mengkaji Model Matematika Pada Teori Eksplosi Termal

Pada tahap ini dilakukan penurunan rumus untuk mendapatkan model matematika pada teori eksplosi termal. Selanjutnya, dilakukan kajian model untuk mendapatkan solusi eksak dan numerik.

3.3 Tahap Mencari Solusi Eksak Model

Pada tahap ini akan dicari solusi eksak dari model matematika pada teori eksplosi termal dengan menggunakan metode dekomposisi Adomian.

3.4 Tahap Mencari Solusi Numerik Model

Pada tahap ini akan dicari solusi numerik dari model matematika pada teori eksplosif termal dengan menggunakan metode deret Taylor.

3.5 Tahap Simulasi Numerik Model Menggunakan *Matlab 2010a*

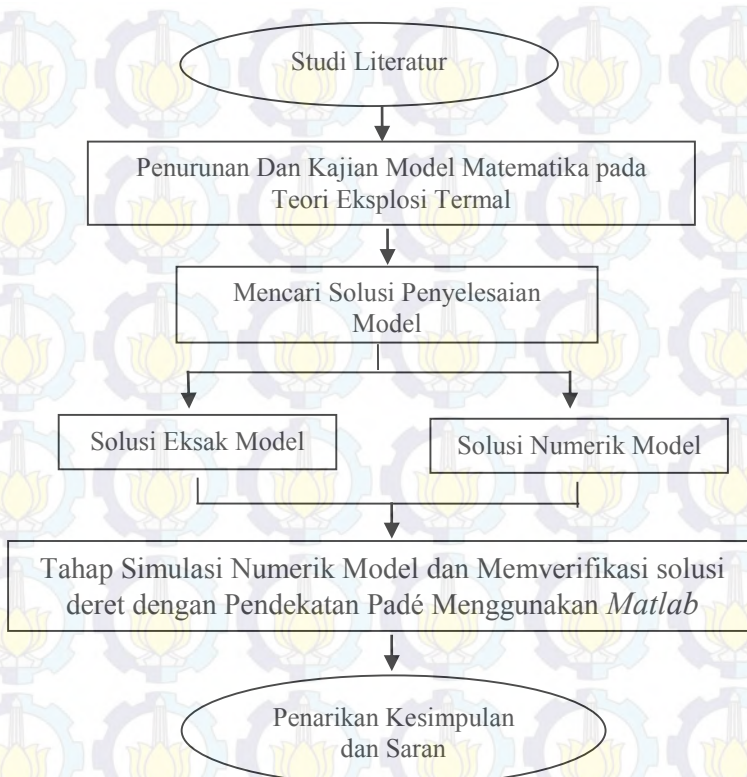
Tahap simulasi numerik model dilakukan menggunakan software pemrograman *Matlab 2010a* untuk menggambarkan grafik solusi/penyelesaian eksak dan solusi/penyelesaian numerik model matematika pada teori eksplosif termal. Dan selanjutnya, menentukan nilai eror maksimum dari model yang diberikan.

3.6 Kesimpulan dan Saran

Setelah dilakukan analisis dan pembahasan maka dapat ditarik suatu kesimpulan dan saran sebagai masukan untuk pengembangan penelitian lebih lanjut.

3.7 Diagram Alir

Diagram alir dimaksudkan untuk memudahkan dalam pengerjaan Tugas Akhir agar lebih sistematis. Diagram alir yang digunakan pada penelitian ini disajikan pada gambar berikut :



Gambar 3.1. Diagram Alir Penelitian



BAB IV

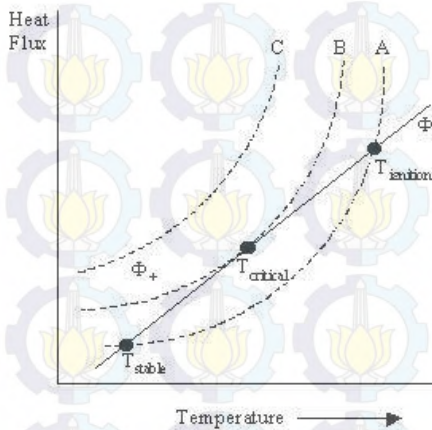
ANALISIS DAN PEMBAHASAN

Pada bab ini akan dibahas definisi dan proses terjadinya eksplosi termal, penurunan model matematika bentuk eksplosi termal, dan solusi penyelesaian dari model matematika eksplosi termal. Pembahasan dimulai dengan penjelasan mengenai keadaan eksplosi termal. Kemudian dilanjutkan dengan penurunan model matematika pada teori eksplosi termal. Selanjutnya, untuk menyelesaikan model matematika pada teori eksplosi termal digunakan metode dekomposisi Adomian dan metode deret Taylor. Setelah itu, pada akhir pembahasan diberikan analisis hasil simulasi untuk menentukan eror maksimum yang bisa dicapai untuk menghindari keadaan eksplosi termal.

4.1 Definisi Dan Proses Terjadinya Eksplosi Termal

Teori ledakan termal didasarkan pada gagasan bahwa pemanasan progresif meningkatkan panas yang dilepaskan oleh reaksi sampai melebihi tingkat kehilangan panas dari sistem. Pada komposisi campuran dan tekanan tertentu yang diberikan, terjadi ledakan panas pada suhu pengapian tertentu yang dapat ditentukan dari perhitungan kehilangan panas dan keuntungan panas [18].

Eksplosi termal adalah istilah yang diambil untuk mengacu pada tahap awal perilaku campuran yang mudah terbakar karena suhu yang mulai meningkat dan bersaing dalam proses fisika dan kimia. Contoh dari keadaan eksplosi termal ini adalah larutan garam (larutan elektrolit).



Gambar 4.1. Plot fluks termal terhadap temperatur

Keterangan gambar :

a Grafik A

Reaktan masuk ke dalam sistem pada suhu rendah dan karena kurva produksi panas berada di atas kurva kehilangan panas akhirnya proses dilanjutkan untuk memanaskan sampai suhu mencapai suhu . Karena suhu ini dikatakan suhu stabil, maka tidak ada pemanasan sendiri selanjutnya akan berlangsung dengan suhu tetap konstan di sekitar suhu yang stabil.

b Grafik B

Ini adalah yang bagian penting dari tiga kurva pada *Gambar 1*. Kurva kehilangan panas adalah tangensial atau kemiringan terhadap kurva mendapatkan panas. Suhu reaktan dalam sistem perlahan akan naik hingga suhu kritis, di mana titik percepatan suhu akan terjadi mengakibatkan ledakan termal.

c Grafik C

Seperti dapat dilihat dari *Gambar 3.1*, kurva fluks mendapatkan panas yang selalu melebihi fluks kehilangan panas. Oleh karena itu pada suhu apa pun reaktan berada dalam sistem, maka ledakan termal akan berlangsung.

4.2 Penurunan Model Matematika Bentuk Eksplosi Termal

Mempertimbangkan sistem nonlinear parabola singular dengan adanya gangguan [19]

$$\varepsilon \frac{\partial \theta}{\partial \tau} = \varphi(\eta) \frac{\theta}{1+\beta\theta} + \frac{1}{\delta} D_{\xi} \theta \quad (4.1)$$

$$\varepsilon \frac{\partial \eta}{\partial \tau} = \varepsilon \varphi(\eta) \frac{\theta}{1+\beta\theta} + \frac{1}{\rho} D_{\xi} \eta \quad (4.2)$$

Dengan

$$D_{\xi}(\cdot) = \frac{\partial^2(\cdot)}{\partial \xi^2} + \frac{n}{\xi} \left(\frac{\partial(\cdot)}{\partial \xi} \right), \quad n = 0,1$$

Tetapi pada model matematika eksplosi termal digunakan $n = 1$ artinya reaktor yang digunakan adalah silinder.

Dengan syarat batas,

$$\left. \frac{\partial \theta}{\partial \xi} \right|_{\xi=0} = 0, \quad \theta|_{\xi=1} = 0, \quad \left. \frac{\partial \eta}{\partial \xi} \right|_{\xi=0} = 0, \quad \left. \frac{\partial \eta}{\partial \xi} \right|_{\xi=1} = 0$$

Dengan syarat awal

$$\theta|_{\tau=1} = 0, \quad \eta|_{\theta=1} = 0$$

Ini adalah model matematika dari eksplosi termal akibat perpindahan panas dan difusi dalam kasus proses pembakaran autokatalitik.

Dengan :

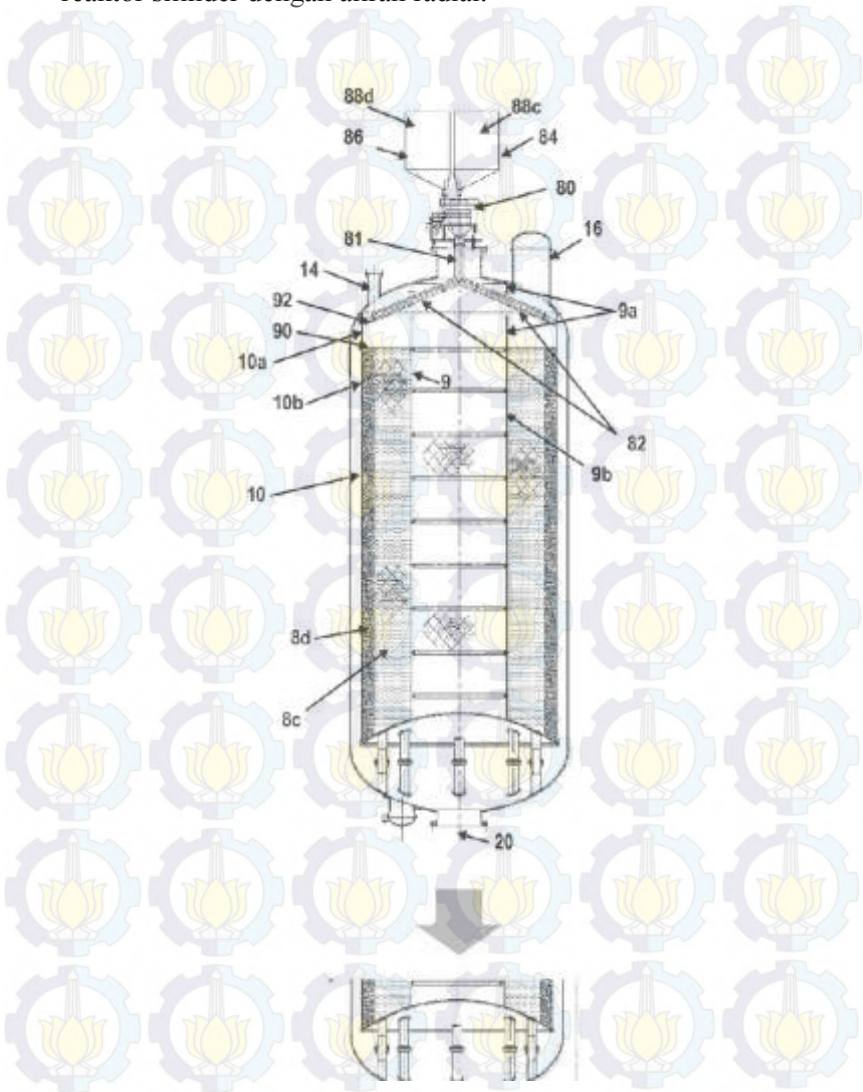
- θ adalah suhu yang tidak berdimensi,
- η adalah tingkat pembakaran yang tidak berdimensi ,
- τ adalah waktu yang tidak berdimensi,
- ε dan β adalah parameter kecil yang positif,
- δ adalah kriteria Frank-Kamenetsky, yaitu parameter skalar, mencirikan keadaan awal dari sistem. Bergantung pada nilai δ , reaksi bisa meledak dan bisa juga berlangsung perlahan-lahan. Nilai dari parameter δ memisahkan reaksi yang lambat dan reaksi yang menyebabkan ledakan yang selanjutnya disebut kondisi kritis.
- Untuk fungsi $\varphi(\eta) = \eta$ diperoleh orde reaksi pertama, untuk $\varphi(\eta) = \eta^n$ adalah orde reaksi ke- n , dan untuk $\varphi(\eta) = \eta(1 - \eta)$ adalah salah satu autokatalitik.

Dengan menggunakan metode teknik integral, nilai kritis dari dihitung δ sebagai deret asimtotik sesuai dengan derajat parameter kecil ε .

$$\delta = \delta_0(1 + \varepsilon\delta_1) + O(\varepsilon^2)$$

Dimana kondisi kritis dimodelkan oleh trayektori *duck*. Untuk $n = 0$ (bidang reaktor paralel) dan untuk $n = 1$ (reaktor silinder) nilai-nilai yang sesuai dari δ_0 dan δ_1 dapat diperkirakan.

Misalnya dibawah ini diberikan gambar bentuk dari reaktor silinder dengan aliran radial.



Gambar 4.2. Reaktor silinder dengan aliran radial

Bentuk parabola di atas mempercepat reaksi kimia dan fisika serta memberikan pancaran satu arah dari larutan yang bereaksi jika terjadi ledakan.

Nilai $n = 1$ pada persamaan (4.1) dan (4.2) dan diselidiki kondisi kritis dari keadaan eksplosi termal pada reaktor silinder. Dengan demikian diperoleh

$$\varepsilon \frac{\partial \theta}{\partial \tau} = \eta(1 - \eta)e^{\theta} + \frac{1}{\delta} \left(\frac{\partial^2 \theta}{\partial \xi^2} + \frac{1}{\xi} \frac{\partial \theta}{\partial \xi} \right) \quad (4.3)$$

$$\varepsilon \frac{\partial \eta}{\partial \tau} = \varepsilon \eta(1 - \eta)e^{\theta} + \frac{1}{\rho} \left(\frac{\partial^2 \eta}{\partial \xi^2} + \frac{1}{\xi} \frac{\partial \eta}{\partial \xi} \right)$$

Dengan syarat batas,

$$\left. \frac{\partial \theta}{\partial \xi} \right|_{\xi=0} = 0, \quad \theta|_{\xi=1} = 0, \quad \left. \frac{\partial \eta}{\partial \xi} \right|_{\xi=0} = 0, \quad \left. \frac{\partial \eta}{\partial \xi} \right|_{\xi=1} = 0$$

Selanjutnya, dicari nilai dari kondisi stabil dan tidak stabil jenis integral satu dimensi dalam bentuk parametrik

$$\theta = \theta(v, \xi, \varepsilon) = \theta_0(v, \xi) + \varepsilon \theta_1(v, \xi) + O(\varepsilon^2)$$

$$\eta = \eta(v, \xi, \varepsilon) = \eta_0(v, \xi) + \varepsilon \eta_1(v, \xi) + O(\varepsilon^2)$$

$$\frac{dv}{dt} = V(v, \xi) = V_0(v) + \varepsilon V_1(v) + O(\varepsilon^2)$$

Faktor δ akan dihitung sebagai ekspansi asimtotik yang berkaitan untuk derajat parameter kecil ε

$$\delta = \delta_0(1 + \varepsilon \delta_1) + O(\varepsilon^2)$$

$$\varepsilon \frac{\partial \theta}{\partial \tau} V = \eta(1 - \eta)e^\theta + \frac{1}{\delta} \left(\frac{\partial^2 \theta}{\partial \xi^2} + \frac{1}{\xi} \frac{\partial \theta}{\partial \xi} \right)$$

$$\varepsilon \frac{\partial \eta}{\partial \tau} V = \varepsilon \eta(1 - \eta)e^\theta + \frac{1}{\varrho} \left(\frac{\partial^2 \eta}{\partial \xi^2} + \frac{1}{\xi} \frac{\partial \eta}{\partial \xi} \right)$$

Dengan pengaturan nilai $\varepsilon = 0$ diperoleh model matematika untuk pendekatan orde nol adalah sebagai berikut

$$\frac{\partial^2 \theta_0}{\partial \xi^2} + \frac{1}{\xi} \frac{\partial \theta_0}{\partial \xi} + \delta_0 \eta_0 (1 - \eta_0) e^{\theta_0} = 0 \quad (4.4)$$

$$\frac{\partial^2 \eta_0}{\partial \xi^2} + \frac{1}{\xi} \frac{\partial \eta_0}{\partial \xi} = 0$$

Dengan syarat batasnya,

$$\left. \frac{\partial \theta_0}{\partial \xi} \right|_{\xi=0} = 0, \quad \theta_0|_{\xi=1} = 0, \quad (4.5)$$

$$\left. \frac{\partial \eta_0}{\partial \xi} \right|_{\xi=0} = 0, \quad \left. \frac{\partial \eta_0}{\partial \xi} \right|_{\xi=1} = 0$$

Fungsi $\eta_0 = \eta_0(v)$ merupakan solusi dari persamaan kedua pada persamaan (4.4) dengan syarat batas pada persamaan (4.5). Dengan v adalah parameter yang bebas, sehingga dapat dituliskan $\eta_0(v, \xi) \equiv v$.

Persamaan pertama pada persamaan (4.4) menjadi

$$\frac{\partial^2 \theta_0}{\partial \xi^2} + \frac{1}{\xi} \frac{\partial \theta_0}{\partial \xi} + \delta_0 v (1 - v) e^{\theta_0} = 0$$

sehingga persamaan di atas dapat ditulis dalam bentuk lain

$$y'' + \frac{1}{\xi} y' = -e^y \quad (4.6)$$

$$y'(0) = 0 \text{ dan } y(0) = 0 \quad (4.7)$$

yang sesuai dengan masalah nilai batas singular BVPS nonlinear timbul dalam teori eksplosi termal/panas [16]

$$y'' + \frac{1}{x} y' = -e^y \quad (4.8)$$

$$y'(0) = 0 \text{ dan } y(0) = 0 \quad (4.9)$$

4.3 Solusi Eksak Model Matematika Pada Teori Eksplosi Termal

Mempertimbangkan masalah nilai batas singular nonlinear pada persamaan (4.8) dan (4.9) maka bentuk umumnya adalah sebagai berikut

$$y'' + \frac{1}{x} y' + v e^y = 0 \quad (4.10)$$

Persamaan ini mempunyai solusi eksak di bawah ini

$$y = 2 \ln \left(\frac{B + 1}{Bx^2 + 1} \right)$$

dengan $B = \frac{(8-2v) \pm \sqrt{(8-2v)^2 - 4v^2}}{2v}$

Sebagai contoh adalah pada persamaan (4.10), ketika $v = 1$ maka solusi eksak yang didapat adalah

$$y = 2 \ln \left(\frac{3 - 2\sqrt{2} + 1}{3 - 2\sqrt{2}x^2 + 1} \right)$$

Jadi, model matematika pada teori eksplosi termal ini sesuai dengan masalah nilai batas singular nonlinear

$$y'' + \frac{1}{x}y' + e^y = 0 \quad (4.11)$$

dengan syarat batas,

$$\lim_{x \rightarrow 0} y(x) = 2 \ln(4 - 2\sqrt{2}) \quad (4.12)$$

$$y(1) = 0$$

Jika menggunakan metode dekomposisi Adomian maka sesuai dengan persamaan (2.9) dan (2.10) maka persamaan (4.11) menjadi

$$(xy')' = -xe^y, \quad 0 < x \leq 1 \quad (4.13)$$

Dengan syarat batas,

$$y_0 = 2 \ln(4 - 2\sqrt{2}) \quad (4.14)$$

$$y_{n+1} = - \int_0^x \left(x^{-1} \int_0^x [xA_n] dx \right) \quad n \geq 0$$

dengan polinomial dari bentuk nonlinear e^y diberikan sebagai berikut

$$\begin{aligned}
 A_0 &= e^{y_0}, & A_1 &= e^{y_0} y_1, & A_2 &= \frac{1}{2} e^{y_0} (y_1^2 + 2y_2) \\
 A_3 &= \frac{1}{6} e^{y_0} (y_1^3 + 6y_1 y_2 + y_3) \\
 A_4 &= \frac{1}{24} e^{y_0} (y_1^4 + 12y_1^2 y_2 + 24y_1 y_3 + 12y_2^2 + 24y_4)
 \end{aligned} \tag{4.15}$$

Dengan menggunakan hubungan perulangan (4.11) dan beberapa bentuk polinomial Adomian pada persamaan (4.15), dapat dengan mudah diperoleh

$$y_0 = 2 \ln(4 - 2\sqrt{2})$$

$$y_1 = -2(3 - 2\sqrt{2})x^2$$

$$y_2 = 2 \left(\frac{(3 - 2\sqrt{2})^2}{2} \right) x^4$$

$$y_3 = -2 \left(\frac{(3 - 2\sqrt{2})^3}{3} \right) x^6$$

$$y_4 = 2 \left(\frac{(3 - 2\sqrt{2})^4}{4} \right) x^8$$

$$y_5 = -2 \left(\frac{(3 - 2\sqrt{2})^5}{5} \right) x^{10}$$

$$y_6 = 2 \left(\frac{(3 - 2\sqrt{2})^6}{6} \right) x^{12}$$

$$y_7 = -2 \left(\frac{(3 - 2\sqrt{2})^2}{2} \right) x^{14}$$

.....

$$y_n = 2 \left(\frac{(-1)^n (3 - 2\sqrt{2})^n}{n} \right) (x^2)^n, \quad n \geq 1$$

Selanjutnya, suku-suku ke- n dapat dideretkan ke dalam bentuk sebagai berikut :

$$y = \sum_{n=0}^{\infty} y_n = y_0 + \sum_{n=1}^{\infty} y_n$$

$$y = 2 \ln(4 - 2\sqrt{2}) + 2 \left(\frac{(-1)^n (3 - 2\sqrt{2})^n}{n} \right) (x^2)^n$$

$$= 2 \ln(4 - 2\sqrt{2}) - 2 \sum_{n=1}^{\infty} \left(\frac{(-1)^{n+1} (3 - 2\sqrt{2})^n}{n} \right) (x^2)^n$$

$$= 2 \ln(4 - 2\sqrt{2}) - 2 \ln[1 + (3 - 2\sqrt{2})] x^2$$

$$= 2 \{ \ln(4 - 2\sqrt{2}) - \ln[1 + (3 - 2\sqrt{2})] x^2 \}$$

$$= 2 \ln \left(\frac{4 - 2\sqrt{2}}{1 + (3 - 2\sqrt{2}) x^2} \right)$$

$$= 2 \ln \left(\frac{3 - 2\sqrt{2} + 1}{(3 - 2\sqrt{2})x^2 + 1} \right) \quad (4.17)$$

Persamaan (4.17) merupakan solusi eksak dari persamaan diferensial pada model matematika eksplosi termal.

4.4 Solusi Numerik Model Matematika Pada Teori Eksplosi Termal Menggunakan Metode Deret Taylor

Untuk menyelesaikan persamaan diferensial yang diberikan pada persamaan (4.8) dan (4.9), Straaus et. al [20] memperkenalkan variabel baru yaitu

$$u(x) = \tanh \left(\frac{y(x)}{4} \right)$$

$$y(x) = 4 \tanh^{-1} u(x) \quad (4.18)$$

$$\frac{dy}{dx} = 4 \left[\frac{1}{1 - u^2} \right] \frac{du}{dx}$$

$$y' = 4 \left[\frac{1}{1 - u^2} \right] u' \quad (4.19)$$

$$y'' = 4 \left[\frac{1}{1 - u^2} \right] u'' + u' \frac{d}{dx} \left[\frac{4}{1 - u^2} \right]$$

$$y'' = \left[\frac{4}{1 - u^2} \right] u'' + \frac{8u}{(1 - u^2)^2} (u')^2 \quad (4.20)$$

Persamaan (4.18), (4.19), dan (4.20) subsitusikan pada persamaan (4.8) dan (4.9) sehingga menjadi

$$\left[\frac{4}{1 - u^2} \right] u'' + \frac{8u}{(1 - u^2)^2} (u')^2 + \frac{1}{x} 4 \left[\frac{1}{1 - u^2} \right] u' = -e^{4 \tanh^{-1} u(x)}$$

$$\left[\frac{4}{1-u^2} \right] u'' + \frac{8u}{(1-u^2)^2} (u')^2 + \frac{1}{x} 4 \left[\frac{1}{1-u^2} \right] u' = - \left(\frac{1+u}{1-u} \right)^2 \cdot \frac{(1-u)^2}{4}$$

Sehingga diperoleh persamaan diferensial baru yaitu

$$\begin{aligned} (1-u)^2 u'' + 2u(u')^2 + \frac{1}{x} (1-u)^2 u' &= -\frac{1}{4} (1+u)^4 \\ (1-u)^2 u'' + 2u(u')^2 + \frac{1}{x} (1-u)^2 u' &= -\frac{1}{4} (1+4u+6u^2+4u^3+u^4) \quad (4.21) \end{aligned}$$

Maka persamaan (4.21) menjadi

$$\begin{aligned} (1-u)^2 x u'' + 2xu(u')^2 + (1-u)^2 u' &= -\frac{1}{4} (1+4u+6u^2+4u^3+u^4) x \\ (1-u)^2 x u'' + 2xu(u')^2 + (1-u)^2 u' &= -\left(\frac{1}{4} + u + \frac{3}{2} u^2 + u^3 + \frac{1}{4} u^4 \right) x \quad (4.22) \end{aligned}$$

Dengan syarat batas,

$$u(1) = 0 \text{ dan } u'(0) = 0 \quad (4.23)$$

Untuk menyelesaikan persamaan diferensial (4.22) dan (4.23) digunakan metode transformasi diferensial. Metode transformasi diferensial ini adalah metode numerik berdasarkan ekspansi Taylor. Metode ini digunakan untuk menemukan koefisien ekspansi serangkaian fungsi yang tidak diketahui dengan menggunakan nilai awal permasalahan. Penyelesaian persamaan diferensial dengan menggunakan metode ini menggunakan perulangan dalam bentuk deret.

Menerapkan operasi transformasi diferensial untuk persamaan (4.22) yang sesuai pada tinjauan pustaka (tabel 2.1), berikut diperoleh hubungan perulangan sebagai berikut

Tabel 4.1 Fungsi Transformasi dari persamaan diferensial pada teori eksplosif termal

Fungsi asli	Fungsi transformasi
$-\frac{1}{4}x$	$-\frac{1}{4}\delta(k-1)$
$-ux$	$-U(k-1)$
$-\frac{3}{2}u^2x$	$-\frac{3}{2}[U(m)\otimes U(k-1)] = -\frac{3}{2}\sum_{m=0}^{k-1}U(m)U(k-m-1)$
$-u^3x$	$-U(k-1)\otimes U(m)\otimes U(n)$ $= \sum_{m=0}^{k-1}U(k-m-1)\sum_{n=0}^mU(m)U(n)$
$-\frac{1}{4}u^4x$	$-\frac{1}{4}[U(k-1)\otimes U(m)\otimes U(n)\otimes U(l)]$ $= -\frac{1}{4}\sum_{m=0}^{k-1}U(k-m$ $-1)\sum_{m=1}^kU(m$ $-l)\sum_{m=1}^lU(n)U(l-n)$
u^2u'	$U(m)\otimes U(n)\otimes(k+1)U(k+1)$ $= \sum_{m=1}^k(k-m+1)U(k-m$ $+1)\sum_{n=0}^mU(n)U(m-n)$

Sehingga persamaan diferensial (4.22) setelah ditransformasi menjadi sebagai berikut :

$$\begin{aligned}
 (k+1)^2[1-U^2(0)]U(k+1) &= -U(k-1) - \frac{1}{4}\delta(k-1) \\
 &+ \sum_{m=1}^{k-1} (k-m)(k-m+1) U(k-m \\
 &+ 1) \sum_{n=0}^m U(n)U(m-n) \\
 &- 2 \sum_{m=0}^{k-1} U(k-m-1) \sum_{n=0}^m (n+1)(m-n \\
 &+ 1) U(n+1) U(m-n+1) \\
 &- \frac{1}{4} \sum_{m=0}^{k-1} U(k-m \\
 &- 1) \sum_{l=0}^m U(m-l) \sum_{n=0}^l U(n)U(l-n) \\
 &+ \sum_{m=1}^k (k-m+1) U(k-m \\
 &+ 1) \sum_{n=0}^m U(n)U(m-n) \\
 &- \frac{3}{2} \sum_{m=0}^{k-1} U(m)U(k-m-1) \\
 &- \sum_{m=0}^{k-1} U(k-m-1) \sum_{n=0}^m U(n)U(m-n), \quad k \\
 &\geq 1
 \end{aligned} \tag{4.24}$$

Hasil transformasi syarat batas berikut ini pada $x = 0$ diperoleh :

$$U(k) = \frac{1}{k!} \frac{d^k u(x)}{dx^k} \Big|_{x=0}$$

Berdasarkan syarat batas pada persamaan (4.23) maka

$$U(1) = \frac{1}{1!} \frac{d u(x)}{dx} \Big|_{x=0}$$

$$U(1) = \frac{1}{1!} u'(0)$$

$$U(1) = 0$$

$$U(0) = \frac{1}{0!} u(0)$$

$$U(0) = u(0) = A$$

Sehingga diperoleh $U(1) = 0$ dan $U(0) = A$. Dengan A adalah sebarang konstanta yang belum diketahui nilainya.

Memecahkan persamaan perulangan (4.24), diperoleh :

Untuk $k = 1$ maka

$$\begin{aligned} U(2) &= \frac{-U(0) - \frac{1}{4} \delta(0) - \frac{1}{4} U^4(0) - \frac{3}{2} U^2(0) - U^3(0)}{4(1 - U^2(0))} \\ &= \frac{(A + 1)^3}{16(A - 1)} \end{aligned}$$

Untuk $k = 2$ maka

$$\begin{aligned}
 (3)^2[1 - U^2(0)]U(3) &= -U(1) - \frac{1}{4}\delta(1) \\
 &+ \sum_{m=1}^{k-1} (k-m)(k-m+1) U(k-m \\
 &+ 1) \sum_{n=0}^m U(n)U(m-n) \\
 &- 2 \sum_{m=0}^{k-1} U(k-m-1) \sum_{n=0}^m (n+1)(m-n \\
 &+ 1) U(n+1) U(m-n+1) \\
 &- \frac{1}{4} \sum_{m=0}^{k-1} U(k-m \\
 &- 1) \sum_{l=0}^m U(m-l) \sum_{n=0}^l U(n)U(l-n) \\
 &+ \sum_{m=1}^k (k-m+1) U(k-m \\
 &+ 1) \sum_{n=0}^m U(n)U(m-n) \\
 &- \frac{3}{2} \sum_{m=0}^{k-1} U(m)U(k-m-1) \\
 &- \sum_{m=0}^{k-1} U(k-m-1) \sum_{n=0}^m U(n)U(m-n)
 \end{aligned}$$

diperoleh $U(3) = 0$

Selanjutnya untuk $k = 3$ maka

$$\begin{aligned}
 (4)^2[1 - U^2(0)]U(4) &= -U(2) - \frac{1}{4}\delta(2) \\
 &+ \sum_{m=1}^{k-1} (k-m)(k-m+1) U(k-m \\
 &+ 1) \sum_{n=0}^m U(n)U(m-n) \\
 &- 2 \sum_{m=0}^{k-1} U(k-m-1) \sum_{n=0}^m (n+1)(m-n \\
 &+ 1) U(n+1) U(m-n+1) \\
 &- \frac{1}{4} \sum_{m=0}^{k-1} U(k-m \\
 &- 1) \sum_{l=0}^m U(m-l) \sum_{n=0}^l U(n)U(l-n) \\
 &+ \sum_{m=1}^k (k-m+1) U(k-m \\
 &+ 1) \sum_{n=0}^m U(n)U(m-n) \\
 &- \frac{3}{2} \sum_{m=0}^{k-1} U(m)U(k-m-1) \\
 &- \sum_{m=0}^{k-1} U(k-m-1) \sum_{n=0}^m U(n)U(m-n)
 \end{aligned}$$

diperoleh $U(4) = \frac{(A+1)^5}{256(A-1)^2}$

Untuk $k = 4$ maka

$$\begin{aligned}
 (5)^2[1 - U^2(0)]U(5) &= -U(3) - \frac{1}{4}\delta(3) \\
 &+ \sum_{m=1}^{k-1} (k-m)(k-m+1) U(k-m \\
 &+ 1) \sum_{n=0}^m U(n)U(m-n) \\
 &- 2 \sum_{m=0}^{k-1} U(k-m-1) \sum_{n=0}^m (n+1)(m-n \\
 &+ 1) U(n+1) U(m-n+1) \\
 &- \frac{1}{4} \sum_{m=0}^{k-1} U(k-m \\
 &- 1) \sum_{l=0}^m U(m-l) \sum_{n=0}^l U(n)U(l-n) \\
 &+ \sum_{m=1}^k (k-m+1) U(k-m \\
 &+ 1) \sum_{n=0}^m U(n)U(m-n) \\
 &- \frac{3}{2} \sum_{m=0}^{k-1} U(m)U(k-m-1) \\
 &- \sum_{m=0}^{k-1} U(k-m-1) \sum_{n=0}^m U(n)U(m-n)
 \end{aligned}$$

diperoleh $U(5) = 0$

Untuk $k = 5$ maka

$$\begin{aligned}
 (6)^2[1 - U^2(0)]U(6) &= -U(4) - \frac{1}{4}\delta(4) \\
 &+ \sum_{m=1}^{k-1} (k-m)(k-m+1) U(k-m \\
 &+ 1) \sum_{n=0}^m U(n)U(m-n) \\
 &- 2 \sum_{m=0}^{k-1} U(k-m-1) \sum_{n=0}^m (n+1)(m-n \\
 &+ 1) U(n+1) U(m-n+1) \\
 &- \frac{1}{4} \sum_{m=0}^{k-1} U(k-m \\
 &- 1) \sum_{l=0}^m U(m-l) \sum_{n=0}^l U(n)U(l-n) \\
 &+ \sum_{m=1}^k (k-m+1) U(k-m \\
 &+ 1) \sum_{n=0}^m U(n)U(m-n) \\
 &- \frac{3}{2} \sum_{m=0}^{k-1} U(m)U(k-m-1) \\
 &- \sum_{m=0}^{k-1} U(k-m-1) \sum_{n=0}^m U(n)U(m-n)
 \end{aligned}$$

diperoleh $U(6) = \frac{(A+1)^7}{4096(A-1)^3}$

Untuk $k = 6$ maka

$$\begin{aligned}
 (7)^2[1 - U^2(0)]U(7) &= -U(4) - \frac{1}{4}\delta(4) \\
 &+ \sum_{m=1}^{k-1} (k-m)(k-m+1) U(k-m \\
 &+ 1) \sum_{n=0}^m U(n)U(m-n) \\
 &- 2 \sum_{m=0}^{k-1} U(k-m-1) \sum_{n=0}^m (n+1)(m-n \\
 &+ 1) U(n+1) U(m-n+1) \\
 &- \frac{1}{4} \sum_{m=0}^{k-1} U(k-m \\
 &- 1) \sum_{l=0}^m U(m-l) \sum_{n=0}^l U(n)U(l-n) \\
 &+ \sum_{m=1}^k (k-m+1) U(k-m \\
 &+ 1) \sum_{n=0}^m U(n)U(m-n) \\
 &- \frac{3}{2} \sum_{m=0}^{k-1} U(m)U(k-m-1) \\
 &- \sum_{m=0}^{k-1} U(k-m-1) \sum_{n=0}^m U(n)U(m-n)
 \end{aligned}$$

diperoleh $U(7) = 0$

Untuk $k = 7$ maka

$$\begin{aligned}
 (8)^2[1 - U^2(0)]U(8) &= -U(6) - \frac{1}{4}\delta(6) \\
 &+ \sum_{m=1}^{k-1} (k-m)(k-m+1) U(k-m \\
 &+ 1) \sum_{n=0}^m U(n)U(m-n) \\
 &- 2 \sum_{m=0}^{k-1} U(k-m-1) \sum_{n=0}^m (n+1)(m-n \\
 &+ 1) U(n+1) U(m-n+1) \\
 &- \frac{1}{4} \sum_{m=0}^{k-1} U(k-m \\
 &- 1) \sum_{l=0}^m U(m-l) \sum_{n=0}^l U(n)U(l-n) \\
 &+ \sum_{m=1}^k (k-m+1) U(k-m \\
 &+ 1) \sum_{n=0}^m U(n)U(m-n) \\
 &- \frac{3}{2} \sum_{m=0}^{k-1} U(m)U(k-m-1) \\
 &- \sum_{m=0}^{k-1} U(k-m-1) \sum_{n=0}^m U(n)U(m-n)
 \end{aligned}$$

diperoleh $U(8) = \frac{(A+1)^9}{65536(A-1)^4}$

Berdasarkan hubungan perulangan sebelumnya diperoleh :

$$U(0) = A$$

$$U(1) = 0$$

$$U(2) = \frac{(A+1)^3}{16(A-1)}$$

$$U(3) = 0$$

$$U(4) = \frac{(A+1)^5}{256(A-1)^2}$$

.

.

.

$$U(k) = \begin{cases} A, & k = 0 \\ 0, & k = \text{ganjil} \\ \frac{(A+1)^{k+1}}{4^k(A-1)^{\frac{k}{2}}}, & k = \text{genap} \end{cases}$$

dengan $k \in \{N+0\}$. Jadi

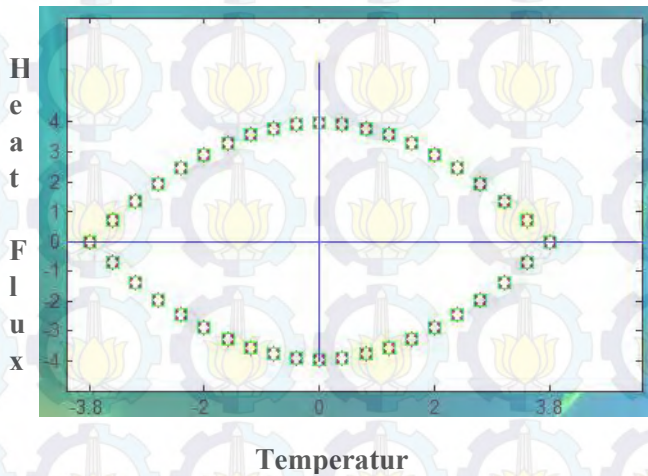
$$u(x)$$

$$\begin{aligned} &= A + 0x + \frac{(A+1)^3}{16(A-1)}x^2 + 0x^3 + \frac{(A+1)^5}{256(A-1)^2}x^4 + 0x^5 \\ &+ \frac{(A+1)^7}{4096(A-1)^3}x^6 + 0x^7 + \frac{(A+1)^9}{65536(A-1)^4}x^8 \\ &+ \dots \end{aligned}$$

Atau lebih singkatnya ditulis

$$\begin{aligned}
 u(x) &= A + \frac{(A+1)^3}{16(A-1)}x^2 + \frac{(A+1)^5}{256(A-1)^2}x^4 + \frac{(A+1)^7}{4096(A-1)^3}x^6 \\
 &+ \frac{(A+1)^9}{65536(A-1)^4}x^8 + \dots
 \end{aligned} \quad (4.25)$$

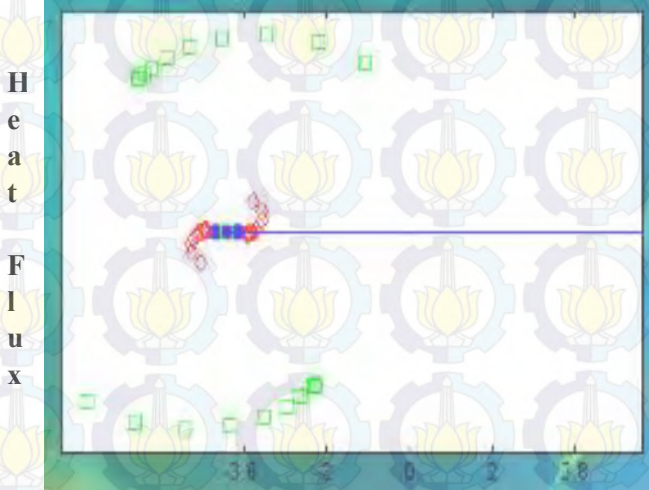
Berikut ditampilkan beberapa simulasi model matematika pada teori eksplosi termal



Gambar 4.3. Singularitas di bidang kompleks x dengan nilai A bernilai 0.07900857355822, 0.07900857355927 dan 0.07900857355927 masing-masing untuk $M=20, 40$, dan 60

Dengan mengambil A bernilai 0.07900857355822, 0.07900857355927 dan 0.07900857355927 masing-masing untuk

$M=20, 40$, dan 60 , maka solusi dari deret pada persamaan (4.25) maka solusi dari deret tersebut akan konvergen ke suatu nilai.



Temperatur

Gambar 4.4. Singularitas di bidang kompleks x dengan nilai A bernilai

- $0.027958791937068318689303963249272 i + 0.74840537855354747157484168762119$
- $0.26488823999378558276401828767185 I + 1.0785841597402998260069158468162$ dan,
- $5.3695481056208137856823282397704 - 13.870916256491247126448395510149 i$
masing-masing untuk $M=20, 40$, dan 60

Dengan mengambil nilai A seperti pada gambar 4.4 maka solusi dari deret pada persamaan (4.25) tidak konvergen ke suatu nilai tertentu.

Sehingga pada A bernilai 0.07900857355822 , 0.07900857355927 dan 0.07900857355927 , untuk $M = 20, 40$ dan

60 dan diverifikasi bahwa solusi deret pada persamaan (4.25) konvergen dalam $x \in (0,1]$ dengan menggunakan pendekatan Padé. Error maksimum untuk $u_M(x)$ tercantum dalam tabel dibawah ini

Tabel 4.3 Maksimum eror yang dicapai

Nilai Max Error	$M = 20$	$M = 40$	$M = 60$
	$2.0018 e - 31$	$2.8979 e - 72$	$1.8103 e - 117$



BAB V PENUTUP

Pada bab ini, diberikan kesimpulan yang diperoleh berdasarkan pembahasan dan hasil simulasi dari software *Matlab 2010a* serta saran untuk penelitian selanjutnya.

5.1 Kesimpulan

Berdasarkan analisis dan pembahasan yang telah disajikan pada bab sebelumnya, dapat disimpulkan beberapa hal sebagai berikut :

- a. Solusi eksak model matematika pada teori eksplosif termal adalah

$$y(x) = 2 \ln \left(\frac{3 - 2\sqrt{2} + 1}{(3 - 2\sqrt{2})x^2 + 1} \right)$$

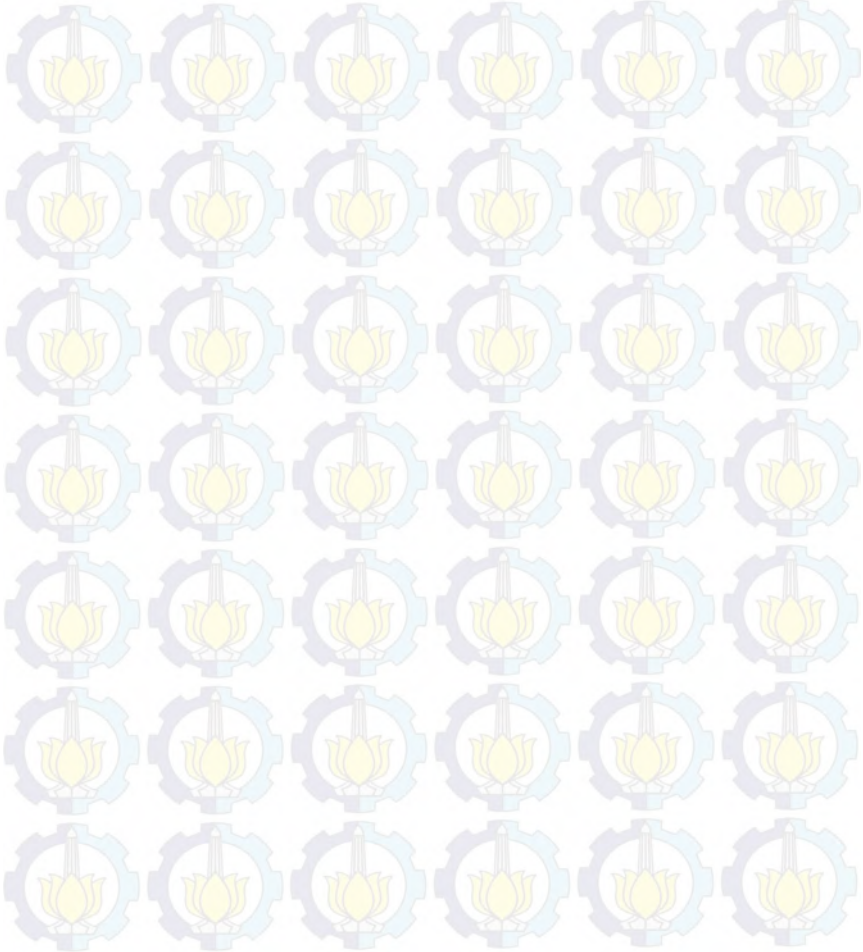
Dan solusi numeriknya adalah

$$\begin{aligned} u(x) = & A + \frac{(A+1)^3}{16(A-1)}x^2 + \frac{(A+1)^5}{256(A-1)^2}x^4 \\ & + \frac{(A+1)^7}{4096(A-1)^3}x^6 + \frac{(A+1)^9}{65536(A-1)^4}x^8 \\ & + O(x)^{10} \end{aligned}$$

- b. Titik-titik stabilitas dari grafik diperoleh ketika A bernilai 0.07900857355822, 0.07900857355927 dan 0.07900857355927.
- c. Nilai eror maksimum terjadi adalah 2.0018e-31, 2.8979e-72, dan 1.8103e-117 masing-masing untuk M bernilai 20, 40, dan 60.

5.2 Saran

Pada Tugas Akhir ini belum dibahas mengenai bagaimana proses kimia dan fisika terjadi pada saat campuran bereaksi. Oleh sebab itu, penulis menyarankan agar penelitian selanjutnya dapat dilanjutkan pada pembahasan tersebut guna kelengkapan buku ini.



LAMPIRAN A

Listing Dan Running Program Mencari Nilai A saat $M = 20$

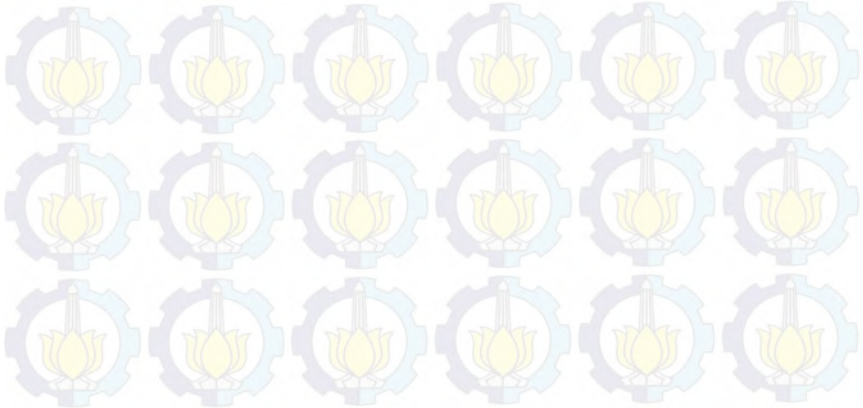
- Listing Program Mencari Nilai A
M-file dengan judul **A1.m**

```
syms a;
solve ('4^20*a*(a - 1)^10 + 4^18*(a + 1)^3*(a - 1)^9 + 4^16*(a - 1)^8*(a + 1)^5 +
4^14*(a - 1)^7*(a + 1)^7 + 4^12*(a - 1)^6*(a + 1)^9 + 4^10*(a - 1)^5*(a + 1)^11 + 4^8*(a - 1)^4*(a + 1)^13 + 4^6*(a - 1)^3*(a + 1)^15 + 4^4*(a - 1)^2*(a + 1)^17 + 4^2*(a - 1)*(a + 1)^19 + (a + 1)^21')
```

- Hasil Running Program

No.	Nilai Faktor dari A
A1	0.079008573558219277924767477956285
A2	12.301172506087224622454480477972 i - 13.443508920474276558086089299216
A3	0.21696159849127427154781861792927 i + 0.89418895613307186778575468224364
A4	8.9232049349730984463882360127649 i + 10.22649539804502872369785524738
A5	0.2675636110983931798371414263825 i + 1.0310193384584077680448878921022
A6	- 16.148662655294506252447663706436 i - 5.2775502213785596770977854717913
A7	0.12927906871940224813994875508305 i + 0.80817690503881058222543790609558
A8	1.2189846510177436975678505249467 - 0.22364638775311280061268287765181 i
A9	0.74840537855354747157484168762119 - 0.027958791937068318689303963249272 i

A10	16.148662655294506252447663706436 i - 5.2775502213785596770977854717913
A11	4.5810990840094228273533508644049 i - 18.30742975469534231922063691073
A12	0.89418895613307186778575468224364 - 0.21696159849127427154781861792927 i
A13	14.905702483003031859701826687354 i + 3.5617139825224588045455000023701
A14	- 4.5810990840094228273533508644049 i - 18.30742975469534231922063691073
A15	1.0310193384584077680448878921022 - 0.2675636110983931798371414263825 i
A16	10.22649539804502872369785524738 - 8.9232049349730984463882360127649 i
A17	3.5617139825224588045455000023701 - 14.905702483003031859701826687354 i
A18	0.80817690503881058222543790609558 - 0.12927906871940224813994875508305 i
A19	- 12.301172506087224622454480477972 i - 13.443508920474276558086089299216
A20	0.22364638775311280061268287765181 i + 1.2189846510177436975678505249467
A21	0.027958791937068318689303963249272 i + 0.74840537855354747157484168762119



LAMPIRAN B

Listing dan running program mencari nilai A saat $M = 40$

- Listing program mencari nilai A
M-file dengan judul **A2.m**

```
syms a;
solve ('4^40*a*(a-1)^20+4^38*(a + 1)^3*(a -
1)^19+4^36*(a + 1)^5*(a - 1)^18+4^34*(a +
1)^7*(a - 1)^17+4^32*(a + 1)^9*(a -
1)^16+4^30*(a + 1)^11*(a - 1)^15+4^28*(a +
1)^13*(a - 1)^14+4^26*(a + 1)^15*(a -
1)^13+4^24*(a + 1)^17*(a - 1)^12+4^22*(a +
1)^19*(a - 1)^11+4^20*(a + 1)^21*(a-1)^10 +
4^18*(a + 1)^23*(a - 1)^9 + 4^16*(a -
1)^8*(a + 1)^25 + 4^14*(a - 1)^7*(a + 1)^27
+ 4^12*(a - 1)^6*(a + 1)^29 + 4^10*(a -
1)^5*(a + 1)^31 + 4^8*(a - 1)^4*(a + 1)^33 +
4^6*(a - 1)^3*(a + 1)^35 + 4^4*(a - 1)^2*(a
+ 1)^37 + 4^2*(a - 1)*(a + 1)^39 + (a +
1)^41')
```

- Hasil Running Program

No	Faktor dari A
A1	0.74601713287414219061214787299466
A2	0.079008573559271753223122088945097
A3	0.77404481171663200010039478638262
A4	6.8449633422910879337997960355872 - 12.809279469644104294963575856906*i

A5	1.1128138145604392950157857100684 - 0.262396201270577070267281342923*i
A6	11.983451892019718617949130213787 - 4.8673266663474125209501815464886*i
A7	1.3035872174103007091588802183731 - 0.13719586423075177142513190635757*i
A8	11.037431503821425312807904606825*i - 14.627568728703837036347626718694
A9	0.11511956509015041745061988745032*i + 0.81286770340201483978581047395803
A10	0.062845534725249586868138054072847*i + 0.78974902652359194722231859656388
A11	7.0369972472833021651851862423076*i - 17.301938835424541764796914599535
A12	1.0231291612539392724702338215809 - 0.26286029817371654917730448565072*i
A13	2.7918121961806091167097170150692 - 15.201998944364245918787324575396*i
A14	0.16329423055613566487897947994077*i + 0.84580132955542820284114989121752
A15	- 11.037431503821425312807904606825*i - 14.627568728703837036347626718694
A16	0.262396201270577070267281342923*i + 1.1128138145604392950157857100684
A17	9.9985223441802855758941140840474 - 9.2658738573934151344684184206449*i
A18	0.89059438683460125276785946979744 - 0.2062830437071009856711787638724*i
A19	15.201998944364245918787324575396*i + 2.7918121961806091167097170150692
A20	16.245577411308615050008118139742*i - 1.7981007606539181638186830477347
A21	0.84580132955542820284114989121752 - 0.16329423055613566487897947994077*i
A22	9.2658738573934151344684184206449*i + 9.9985223441802855758941140840474
A23	- 16.245577411308615050008118139742*i - 1.7981007606539181638186830477347

A24	0.94913756497477009102801294819849 - 0.24120082561046778699236953874525*i
A25	- 14.065535106191930868905645884083*i - 10.919137205154610805337690625764
A26	4.8673266663474125209501815464886*i + 11.983451892019718617949130213787
A27	0.2062830437071009856711787638724*i + 0.89059438683460125276785946979744
A28	0.13719586423075177142513190635757*i + 1.3035872174103007091588802183731
A29	15.854394578856458439589046809244*i - 6.5102235466661372507197299907335
A30	14.065535106191930868905645884083*i - 10.919137205154610805337690625764
A31	12.809279469644104294963575856906*i + 6.8449633422910879337997960355872
A32	- 2.4166332834251261804405308032129*i - 18.701972592986669899822557971674
A33	- 7.0369972472833021651851862423076*i - 17.301938835424541764796914599535
A34	0.78974902652359194722231859656388 - 0.062845534725249586868138054072847*i
A35	0.24120082561046778699236953874525*i + 0.94913756497477009102801294819849
A36	0.81286770340201483978581047395803 - 0.11511956509015041745061988745032*i
A37	0.26286029817371654917730448565072*i + 1.0231291612539392724702338215809
A38	- 15.854394578856458439589046809244*i - 6.5102235466661372507197299907335
A39	2.4166332834251261804405308032129*i - 18.701972592986669899822557971674
A40	0.22582565121845408771541176365346*i + 1.2129764313279050942325621017253
A41	1.2129764313279050942325621017253 - 0.22582565121845408771541176365346*i



LAMPIRAN C

Listing dan running program mencari nilai A saat $M = 60$

- Listing program mencari nilai A
M-file dengan judul **A3.m**

```
syms a;
solve ('4^60*a*(a - 1)^30+4^58*(a + 1)^3*(a - 1)^29+4^56*(a + 1)^5*(a - 1)^28+4^54*(a + 1)^7*(a - 1)^27+4^52*(a + 1)^9*(a - 1)^26+4^50*(a + 1)^11*(a - 1)^25+4^48*(a + 1)^13*(a - 1)^24+4^46*(a + 1)^15*(a - 1)^23+4^44*(a + 1)^17*(a - 1)^22+4^42*(a + 1)^19*(a - 1)^21+4^40*(a + 1)^21*(a-1)^20 + 4^38*(a + 1)^23*(a - 1)^19 + 4^36*(a - 1)^18*(a + 1)^25 + 4^34*(a - 1)^17*(a + 1)^27 + 4^32*(a - 1)^16*(a + 1)^29 + 4^30*(a - 1)^15*(a + 1)^31 + 4^28*(a - 1)^14*(a + 1)^33 + 4^26*(a - 1)^13*(a + 1)^35 + 4^24*(a - 1)^12*(a + 1)^37 + 4^22*(a - 1)^11*(a + 1)^39+4^20*(a - 1)^10*(a + 1)^41+4^18*(a - 1)^9*(a + 1)^43+4^16*(a - 1)^8*(a + 1)^45+4^14*(a - 1)^7*(a + 1)^47+4^12*(a - 1)^6*(a + 1)^49+4^10*(a - 1)^5*(a + 1)^51+4^8*(a - 1)^4*(a + 1)^53+4^6*(a - 1)^3*(a + 1)^55+4^4*(a - 1)^2*(a + 1)^57+4^2*(a - 1)*(a + 1)^59+(a+1)^61')
```

- Hasil Running Program

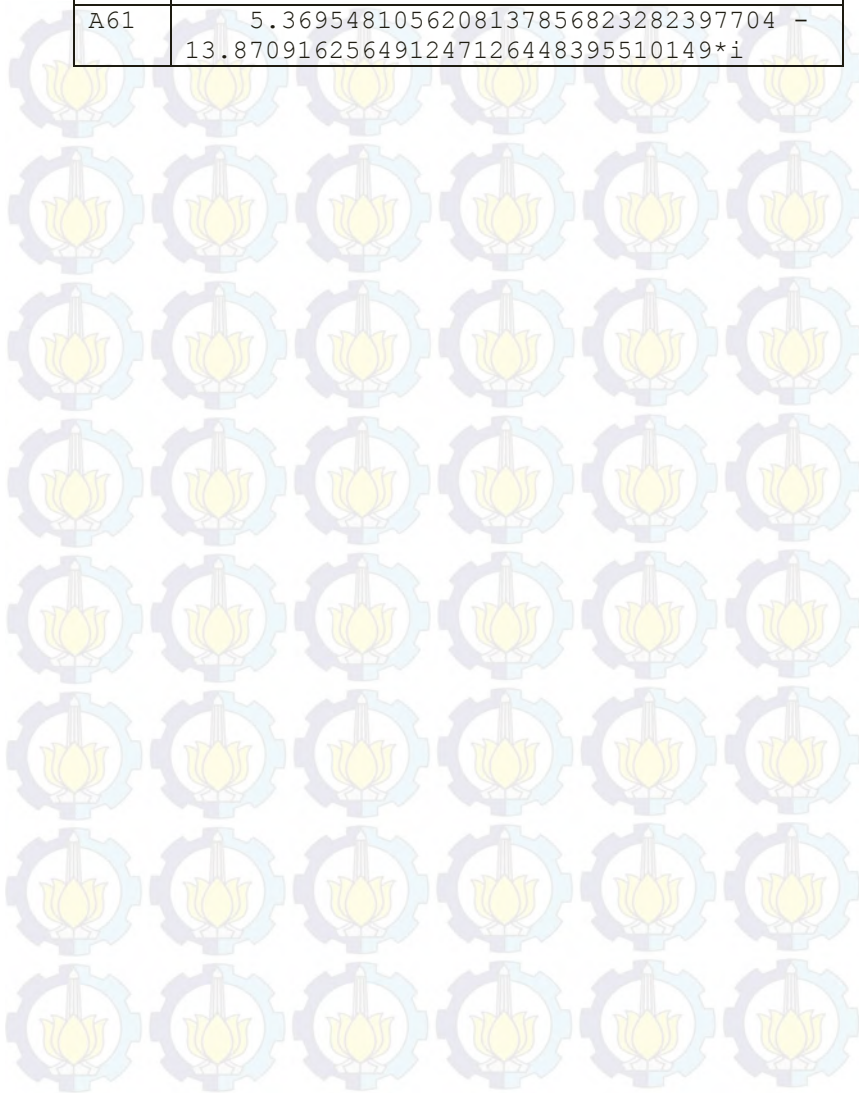
No	Faktor dari A
A1	0.079008573559271753223132064891836
A2	0.74458879245270894626146337404161

A3	0.78391153765195282505508494099117
A4	7.8638861132474686989392823380049*i - 16.85316024859628853078129900227
A5	0.8896231853824342373217179159131 - 0.2026626878831967180410357916635*i
A6	0.22762585780074588158785270481344*i + 0.92631143216645056726030610586953
A7	0.040193825761299627679129489287886*i + 0.78933514665532952545832853168762
A8	- 7.8638861132474686989392823380049*i - 16.85316024859628853078129900227
A9	3.323453064541170583610217601516*i + 12.347423451418393844823461250955
A10	0.26116039189631788899801718057548*i + 1.0205439909334893540642001802836
A11	0.22651100224301025822856895618885*i + 1.2109032566831972035309120617673
A12	0.78933514665532952545832853168762 - 0.040193825761299627679129489287886*i
A13	15.296039942646855100781642278488*i + 2.5151157881811972311149089423897
A14	0.17427945064962478320723985437214*i + 0.8591849252974600951575817813418
A15	16.240724671014695512133379423313*i - 3.7710295922361440782858819374343
A16	0.096180687413181899538474163108996*i + 1.3243931842790472821680174997092
A17	13.870916256491247126448395510149*i + 5.3695481056208137856823282397704
A18	16.096116040788566137529283685573*i - 0.57461302448597262912456672558516
A19	0.2026626878831967180410357916635*i + 0.8896231853824342373217179159131
A20	0.81487823605712441130450759436033 - 0.11068269482464420885407563316666*i
A21	4.8497524037161444858695848155548*i - 18.123323412213844473889574778755

A22	1.3243931842790472821680174997092 - 0.096180687413181899538474163108996*i
A23	1.2754045679089039994895866824242 - 0.17417587843838043860014147859557*i
A24	- 4.8497524037161444858695848155548*i - 18.123323412213844473889574778755
A25	12.825170917500825227083199625585*i - 12.681599418982142456142389558526
A26	0.24778992151461113671154358782436*i + 0.96980498219394887534315594872443
A27	14.570934280407889542919760754275*i - 9.9529502791051832903616329053113
A28	- 1.638758002490539763376794974632*i - 18.771815350331261627080205413904
A29	9.3853045218555149938049084210198*i + 9.9155057094217959625534215659905
A30	- 10.55881923979378002626387848089*i - 15.013874242492766290293877078294
A31	1.1430256296555456402586268513018 - 0.25502392790699374977766002465987*i
A32	0.17417587843838043860014147859557*i + 1.2754045679089039994895866824242
A33	1.2109032566831972035309120617673 - 0.22651100224301025822856895618885*i
A34	2.5151157881811972311149089423897 - 15.296039942646855100781642278488*i
A35	- 15.725157027169546068375471070958*i - 6.9410764462471994336199042060298
A36	9.9155057094217959625534215659905 - 9.3853045218555149938049084210198*i
A37	0.14341644275482190585893281146312*i + 0.83444217195925483665138854190301
A38	- 14.570934280407889542919760754275*i - 9.9529502791051832903616329053113
A39	1.0205439909334893540642001802836 - 0.26116039189631788899801718057548*i
A40	0.8591849252974600951575817813418 - 0.17427945064962478320723985437214*i

A41	0.26488823999378558276401828767185*i + 1.0785841597402998260069158468162
A42	- 12.825170917500825227083199625585*i - 12.681599418982142456142389558526
A43	6.496612441725059834981776248257*i + 11.425296858383900595491327804946
A44	- 16.240724671014695512133379423313*i - 3.7710295922361440782858819374343
A45	0.076370433260209259615864497889186*i + 0.80000250993646730601661670735436
A46	0.83444217195925483665138854190301 - 0.14341644275482190585893281146312*i
A47	11.425296858383900595491327804946 - 6.496612441725059834981776248257*i
A48	7.8703602709837814676121813626388 - 11.875265024374184521901937056842*i
A49	15.725157027169546068375471070958*i - 6.9410764462471994336199042060298
A50	0.80000250993646730601661670735436 - 0.076370433260209259615864497889186*i
A51	1.638758002490539763376794974632*i - 18.771815350331261627080205413904
A52	- 16.096116040788566137529283685573*i - 0.57461302448597262912456672558516
A53	1.0785841597402998260069158468162 - 0.26488823999378558276401828767185*i
A54	0.11068269482464420885407563316666*i + 0.81487823605712441130450759436033
A55	11.875265024374184521901937056842*i + 7.8703602709837814676121813626388
A56	12.347423451418393844823461250955 - 3.323453064541170583610217601516*i
A57	10.55881923979378002626387848089*i - 15.013874242492766290293877078294
A58	0.25502392790699374977766002465987*i + 1.1430256296555456402586268513018
A59	0.96980498219394887534315594872443 - 0.24778992151461113671154358782436*i

A60	$0.92631143216645056726030610586953 - 0.22762585780074588158785270481344*i$
A61	$5.3695481056208137856823282397704 - 13.870916256491247126448395510149*i$





LAMPIRAN D

- Desain GUI untuk simulasi
 - Tampilan awal



Beberapa fasilitas yang disediakan yaitu :

- a *Home*, masuk pada tampilan utama
- b *Keluar*, keluar dari program.

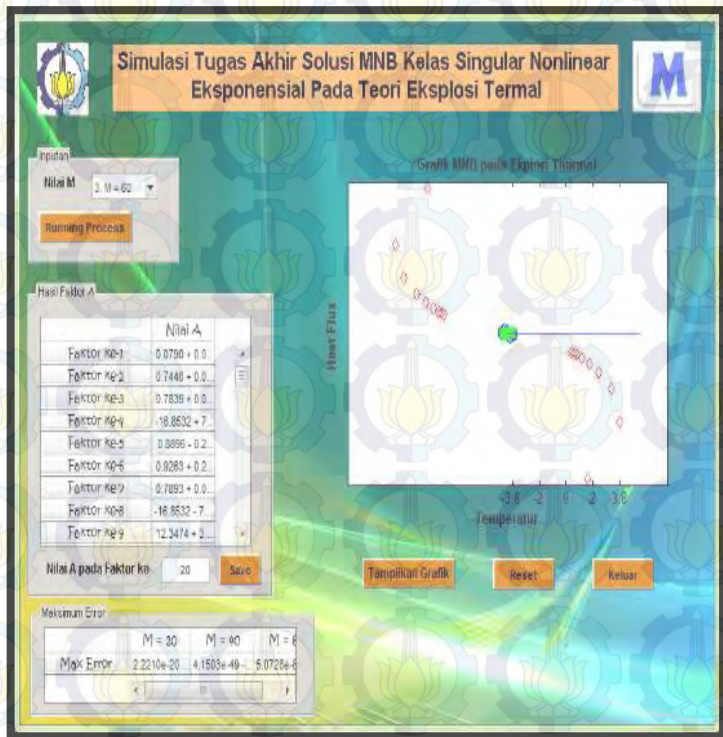
➤ Tampilan utama



Terdapat beberapa fasilitas yang disediakan yaitu :

- Input nilai M* , pada pop up menu M disediakan pilihan $M = 20$, $M = 40$, dan $M = 60$
- Running process*, merupakan tombol eksekusi untuk mencari nilai A .
- Tabel faktor A* , menampilkan semua nilai dari A .
- Nilai A pada Faktor ke-*, artinya adanya faktor beberapa dari A yang ingin user inputkan.
- Save*, artinya nilai dari A pada faktor ke- akan disimpan untuk dieksekusi.

- f *Tabel maksimum eror*, menampilkan hasil eksekusi perintah dengan menampilkan nilai maksimum eror.
 - g *Grafik*, menampilkan simulasi grafik.
 - h *Reset*, artinya menghapus perintah yang pernah dilakukan sebelumnya.
 - i *Keluar*, artinya keluar dari tampilan utama.
- Contoh hasil simulasi running (simulasi) dengan menggunakan GUI





LAMPIRAN E

Listing program mencari nilai maksimum eror dan simulasi grafik

- Listing Program M-file dengan judul **Solusi_MNB_Singular.m**

```
function varargout =
Solusi_MNB_Singular(varargin)
% SOLUSI_MNB_SINGULAR M-file for
Solusi_MNB_Singular.fig
%     SOLUSI_MNB_SINGULAR, by itself,
creates a new SOLUSI_MNB_SINGULAR or
raises the existing
%     singleton*.
%
%     H = SOLUSI_MNB_SINGULAR returns
the handle to a new
SOLUSI_MNB_SINGULAR or the handle to
%     the existing singleton*.
%
%
%     SOLUSI_MNB_SINGULAR('CALLBACK', hObject
,eventData,handles,...) calls the
local
%     function named CALLBACK in
SOLUSI_MNB_SINGULAR.M with the given
input arguments.
%
%
%     SOLUSI_MNB_SINGULAR('Property','Value'
,...) creates a new
SOLUSI_MNB_SINGULAR or raises the
%     existing singleton*. Starting
from the left, property value pairs
are
```



```

%      applied to the GUI before
Solusi_MNB_Singular_OpeningFcn gets
called. An
%      unrecognized property name or
invalid value makes property
application
%      stop. All inputs are passed to
Solusi_MNB_Singular_OpeningFcn via
varargin.
%
%      *See GUI Options on GUIDE's
Tools menu. Choose "GUI allows only
one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
% Edit the above text to modify the
response to help Solusi_MNB_Singular
% Last Modified by GUIDE v2.5 27-Jan-
2015 12:59:57

% Begin initialization code - DO NOT
EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',
mfilename, ...
                  'gui_Singleton',
gui_Singleton, ...
                  'gui_OpeningFcn',
@Solusi_MNB_Singular_OpeningFcn, ...
                  'gui_OutputFcn',
@Solusi_MNB_Singular_OutputFcn, ...
                  'gui_LayoutFcn',
[] , ...

```

```

                                'gui_Callback',
    []);
    if nargin && ischar(varargin{1})
        gui_State.gui_Callback =
            str2func(varargin{1});
    end

    if nargout
        [varargout{1:nargout}] =
            gui_mainfcn(gui_State, varargin{:});
    else
        gui_mainfcn(gui_State,
            varargin{:});
    end
    % End initialization code - DO NOT
    EDIT

    % --- Executes just before
    Solusi_MNB_Singular is made visible.
    function
    Solusi_MNB_Singular_OpeningFcn(hObject
    , eventdata, handles, varargin)
    % This function has no output args,
    see OutputFcn.
    % hObject    handle to figure
    % eventdata  reserved - to be defined
    in a future version of MATLAB
    % handles    structure with handles
    and user data (see GUIDATA)
    % varargin   command line arguments to
    Solusi_MNB_Singular (see VARARGIN)

    % Choose default command line output
    for Solusi_MNB_Singular
    handles.output = hObject;

```

```

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Solusi_MNB_Singular
wait for user response (see UIRESUME)
% uiwait(handles.figure1);

handles.gambar=imread('ITS.jpg');
axes(handles.axes2);
imshow(handles.gambar);

handles.gambar2=imread('Matematika.jpg
');
axes(handles.axes3);
imshow(handles.gambar2);

background =
axes('unit','normalized','position',[0
0 1 1]);
cover=imread('img36.jpg');imagesc(cove
r);
set(background,'handlevisibility','off
','visible','off');
uistack(background,'bottom');

% --- Outputs from this function are
returned to the command line.
function varargout =
Solusi_MNB_Singular_OutputFcn(hObject,
eventdata, handles)
% varargout    cell array for returning
output args (see VARARGOUT);
% hObject     handle to figure
% eventdata   reserved - to be defined
in a future version of MATLAB

```

```
% handles      structure with handles
and user data (see GUIDATA)

% Get default command line output from
handles structure
varargout{1} = handles.output;

function
Hasil_Grafik_edit_Callback(hObject,
eventdata, handles)
% hObject      handle to
Hasil_Grafik_edit (see GCBO)
% eventdata    reserved - to be defined
in a future version of MATLAB
% handles      structure with handles
and user data (see GUIDATA)

% Hints: get(hObject,'String') returns
contents of Hasil_Grafik_edit as text
%
str2double(get(hObject,'String'))
returns contents of Hasil_Grafik_edit
as a double

% --- Executes during object creation,
after setting all properties.
function
Hasil_Grafik_edit_CreateFcn(hObject,
eventdata, handles)
% hObject      handle to
Hasil_Grafik_edit (see GCBO)
% eventdata    reserved - to be defined
in a future version of MATLAB
```



```

% handles      empty - handles not
created until after all CreateFcns
called

% Hint: edit controls usually have a
white background on Windows.
%      See ISPC and COMPUTER.
if ispc &&
isequal(get(hObject,'BackgroundColor')
,
get(0,'defaultUicontrolBackgroundColor')
)
set(hObject,'BackgroundColor','white')
;
end

% --- Executes on button press in
Grafik_pushbutton.
function
Grafik_pushbutton_Callback(hObject,
eventdata, handles)
% hObject      handle to
Grafik_pushbutton (see GCBO)
% eventdata    reserved - to be defined
in a future version of MATLAB
% handles      structure with handles
and user data (see GUIDATA)
A1 = handles.A1;
A2 = handles.A2;
A3 = handles.A3;
A = [A1 A2 A3];

% A(1) = 0.07900857355822;
% A(2) = 0.07900857355927;

```

```

% A(3) = 0.07900857355927;
M(1) = 20;
M(2) = 40;
M(3) = 60;

k=0;
for x=-1:.1:1
    k=k+1;
    [Jumlah11 Jumlah12 K(1)] =
    Jumlahan_BVP(A(1),M(1),x);
    u11(k) = Jumlah11;
    u12(k) = Jumlah12;
    [Jumlah21 Jumlah22 K(2)] =
    Jumlahan_BVP(A(2),M(2),x);
    u21(k) = Jumlah21;
    u22(k) = Jumlah22;
    [Jumlah31 Jumlah32 K(3)] =
    Jumlahan_BVP(A(3),M(3),x);
    u31(k) = Jumlah31;
    u32(k) = Jumlah32;
end

axes(handles.axes1);
x1 = 0:.1:25;
y1 = zeros(1, (25/.1)+1);
x2 = 11*ones(1, (.24/.1)+1);
y2 = -0.08:0.1:0.16;
plot(u11,'o');
hold on;
plot(u12,'o');
% hold on;
plot(u21,'rd');
plot(u22,'rd');
plot(u31,'gs');
plot(u32,'gs');
plot(x1,y1,'-');

```

```

plot(x2,y2,'-');
title('Grafik MNB pada Eksplosi
Thermal','fontweight','b');
xlabel('Temperatur','fontweight','b');
ylabel('Heat Flux','fontweight','b');
set(gca,'XTick',1:5:21)
set(gca,'XTickLabel',{'-3.8','-
2','0','2','3.8'})
set(gca,'YTick',-0.08:0.02:0.08)
set(gca,'YTickLabel',{'-4','-3','-
2','-1','0','1','2','3','4'})

for i=1:length(M)
    Error(i) = K(i)/factorial(M(i)+1);
end
Kolom = {'M = 20','M = 40','M = 60'};
Baris = {'Max Error'};

set(handles.Max_Error_uitable,'data',E
rror,'RowName',Baris,'ColumnName',Kolo
m);

% --- Executes on button press in
Reset_pushbutton.
function
Reset_pushbutton_Callback(hObject,
eventdata, handles)
% hObject      handle to
Reset_pushbutton (see GCBO)
% eventdata    reserved - to be defined
in a future version of MATLAB
% handles      structure with handles
and user data (see GUIDATA)

set(handles.A_edit,'String','');

```

```

set(handles.Max_Error_uitable,'data','
','RowName','','ColumnName','');
set(handles.Hasil_uitable,'data','','RowName','','ColumnName','');

axes(handles.axes1);
plot(0);
hold off;

% --- Executes on button press in
Keluar_pushbutton.
function
Keluar_pushbutton_Callback(hObject,
eventdata, handles)
% hObject    handle to Keluar_pushbutton
% (see GCBO)
% eventdata  reserved - to be defined in
a future version of MATLAB
% handles    structure with handles and
user data (see GUIDATA)

close

function A_edit_Callback(hObject,
eventdata, handles)
% hObject    handle to A_edit (see GCBO)
% eventdata  reserved - to be defined in
a future version of MATLAB
% handles    structure with handles and
user data (see GUIDATA)

% Hints: get(hObject,'String') returns
contents of A_edit as text
%
str2double(get(hObject,'String')) returns
contents of A_edit as a double

```



```

% --- Executes during object creation,
after setting all properties.
function A_edit_CreateFcn(hObject,
eventdata, handles)
% hObject    handle to A_edit (see GCBO)
% eventdata  reserved - to be defined in
a future version of MATLAB
% handles    empty - handles not created
until after all CreateFcns called

% Hint: edit controls usually have a
white background on Windows.
%         See ISPC and COMPUTER.
if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))

set(hObject,'BackgroundColor','white');
end

% --- Executes on selection change in
M_popupmenu.
function M_popupmenu_Callback(hObject,
eventdata, handles)
% hObject    handle to M_popupmenu (see
GCBO)
% eventdata  reserved - to be defined in
a future version of MATLAB
% handles    structure with handles and
user data (see GUIDATA)

% Hints: contents =
cellstr(get(hObject,'String')) returns
M_popupmenu contents as cell array

```

```

%           contents{get(hObject,'Value')}
returns selected item from M_popupmenu

% --- Executes during object creation,
after setting all properties.
function M_popupmenu_CreateFcn(hObject,
 eventdata, handles)
% hObject    handle to M_popupmenu (see
GCBO)
% eventdata  reserved - to be defined in
a future version of MATLAB
% handles    empty - handles not created
until after all CreateFcns called

% Hint: popupmenu controls usually have a
white background on Windows.
%           See ISPC and COMPUTER.
if ispc &&
isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in
Running_Process_pushbutton.
function
Running_Process_pushbutton_Callback(hObject,
 eventdata, handles)
% hObject    handle to
Running_Process_pushbutton (see GCBO)
% eventdata  reserved - to be defined in
a future version of MATLAB

```

```
% handles      structure with handles and
user data (see GUIDATA)
```

```
Pilihan_Nilai_M =
get(handles.M_popupmenu, 'value');
```

```
switch Pilihan_Nilai_M
```

```
case 1
```

```
    S = A1();
```

```
    H1 = double(S);
```

```
    Kolom = {'Nilai A'};
```

```
    for i=1:21
```

```
        Baris(i) = {sprintf('Faktor
ke-%d', i)};
    end
```

```
set(handles.Hasil_uitable, 'data', H1, 'RowName', Baris, 'ColumnName', Kolom);
```

```
handles.H1=H1;
```

```
case 2
```

```
    S = A2();
```

```
    H2 = double(S);
```

```
    Kolom = {'Nilai A'};
```

```
    for i=1:21
```

```
        Baris(i) = {sprintf('Faktor
ke-%d', i)};
    end
```

```
set(handles.Hasil_uitable, 'data', H2, 'RowName', Baris, 'ColumnName', Kolom);
```

```
handles.H2=H2;
```

```
case 3
```

```
    S = A3();
```

```
    H3 = double(S);
```

```
    Kolom = {'Nilai A'};
```

```
    for i=1:21
```

```

        Baris(i) = {sprintf('Faktor
ke-%d',i)};
    end

    set(handles.Hasil_uitable,'data',H3,'RowName',Baris,'ColumnName',Kolom);
    handles.H3=H3;
end
guidata(hObject, handles);

% --- Executes on button press in Save_pushbutton.
function
Save_pushbutton_Callback(hObject, eventdata, handles)
% hObject      handle to Save_pushbutton
(see GCBO)
% eventdata    reserved - to be defined in
a future version of MATLAB
% handles      structure with handles and
user data (see GUIDATA)
Faktor =
str2double(get(handles.A_edit,'String'));
Pilihan_Nilai_M =
get(handles.M_popupmenu,'value');

switch Pilihan_Nilai_M
case 1
    H1 = handles.H1;
    A1 =  H1(Faktor);
    handles.A1=A1;
case 2
    H2 = handles.H2;
    A2 =  H2(Faktor);
    handles.A2=A2;

```



```
case 3
    H3 = handles.H3;
    A3 = H3(Faktor);
    handles.A3=A3;
end
guidata(hObject, handles);
```